From Machine Learning to Heterogeneous Models: A Comprehensive Study on Fine-Grained Emotion Detection in Arabic Text

Esra'a Alshdaifat
Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa, Jordan
esraa@hu.edu.jo (corresponding author)

Enshirah Altarawneh
Department of Computer Engineering, Faculty of Engineering, The Hashemite University, Zarqa, Jordan
enshirah@hu.edu.jo

Mogeeb Alrahman Aloshaibat
Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa, Jordan
Mojeebhasan69@gmail.com

Anas Hussein
Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa, Jordan
a.m.s.ai.ml.ds@gmail.com

Mohammed Dawood
Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa, Jordan
mhmddawwd1231@gmail.com

Tawfiq Tahaineh
Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa, Jordan
tawfiq2002@yahoo.com

Received: 22 January 2026 | Revised: 24 February 2026 | Accepted: 4 March 2026

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | *DOI: https://doi.org/10.48084/etasr.17705*

ABSTRACT

Recognizing emotions from Arabic text has received increasing attention in the last few years, due to its significant role in several real-world applications. This study begins by employing several classic Machine Learning (ML), Deep Learning (DL), and Pre-trained Language Models (PLMs) to construct emotion detection models for Arabic text. Then, a composite model is proposed that integrates the most-effective model from each considered category. More precisely, two main strategies are examined to combine the most-effective models: (i) a parallel strategy and (ii) a sequential strategy. Adopting the parallel strategy, the base models are independent and combined utilizing majority voting or average probability techniques, whereas in the sequential strategy the models are dependent and different weights are assigned to the base models. According to the extensive experiments conducted, the best F1-scores obtained from ML, DL, and PLMs were 0.668, 0.742, and 0.794, respectively. On the other hand, the sequential composite model produced an F1-score of 0.802, outperforming the parallel models, which reached an F1-score of 0.790. The findings highlight the power of sequential hybrid models and confirm that integrating different

architectural models in a way that one model passes information to the succeeding model, focusing on learning hard instances, outperforms all considered models.

Keywords-emotion detection; Natural Language Processing (NLP); heterogeneous models; sequential hybrid models

I. INTRODUCTION AND BACKGROUND

Recently, several applications rely on detecting emotions from text to achieve their intended goals, such as social media analytics, mental health chatbots, and customer feedback analysis. Thus, developing an identification system to recognize different emotions has become crucial in the discipline of Natural Language Processing (NLP). According to the level of granularity, emotion detection can be classified into two major classes: coarse-grained and fine-grained approaches. In coarse-grained emotion detection, the text is classified into a maximum of three class labels (positive, negative, and neutral). For example, the feedback "I love this hotel!" is classified as "positive," whereas "the service was terrible" is assigned a negative label. In fine-grained analysis, the process is more challenging due to the inclusion of several specific emotions, in particular fear, anger, joy, sadness, surprise, and disgust. For instance, in a mental health chatbot context, the Arabic sentence "أشعر بالوحدة والحزن ولا أحد يفهمني" ("I feel lonely and sad and no one understands me") is classified as "sadness," not as a negative sentiment, such as in the case of coarse-grained emotion detection. From the foregoing, fine-grained emotion detection is considered a multiclass classification problem.

With respect to English, significant research has been conducted to develop fine-grained emotion detection. In contrast, the work directed at the Arabic language, which is considered one of the most universally spoken languages, is still underexplored. More precisely, most research work in Arabic NLP has focused on coarse-grained emotion detection [1]. Fine-grained emotion detection is less covered, and the main reasons are the lack of high-quality datasets and the challenges related to multiclass classification. From the literature, four main approaches have been utilized for emotion detection from Arabic text: (i) lexicon-based approaches, (ii) conventional Machine Learning (ML) approaches, (iii) Deep Learning (DL) approaches, and (iv) recent Pre-trained Language Models (PLMs).

Commencing with the simplest approach for recognizing emotions from textual data, the lexicon-based approach uses pre-defined "emotion lexicons" to determine the emotion in a text. More specifically, for identifying the emotion in a given sentence, its words are compared with the lexicon representing each emotion. The emotion with the largest match score is assigned to the sentence. In addition to its simplicity, lexicon-based methods are interpretable and have low computational cost [2]. However, the Arabic language is known for its rich morphology and dialectal diversity. Thus, building a robust lexicon for each Arabic emotion is a challenging task [3, 4]. Despite their effectiveness in discovering essential emotional signals, lexicon-based methods are ineffective in capturing context-dependent meanings [5]. Due to the weaknesses associated with lexicon-based methods, researchers focused on creating prediction models that are able to predict the correct emotion from a given text. The process starts with representing the dataset using text representation techniques such as: (i) Bag of Words (BoW), (ii) lexicons, (iii) n-grams, (iv) Term Frequency–Inverse Document Frequency (TF–IDF), and (v) text embeddings. Then, the encoded dataset is fed into the ML algorithm to build the intended prediction model. A comprehensive study was conducted by authors in [6], in which they utilized the SemEval 2018 dataset [1], and performed several prediction tasks, including emotion intensity regression, emotion intensity ordinal classification, and emotion classification. Several text representation techniques, including lexicons, n-grams, AraVec embeddings [7], and FastText embeddings [8] were employed. These were coupled with several classification algorithms: Ridge Classifier (RC), Random Forest (RF), Support Vector Machines (SVM), and ensemble classification. According to the reported results, the best classifier was task-dependent, and word embeddings generated the most promising outcomes compared to other text representation techniques.

For the same dataset, the SemEval-2018 dataset, authors in [9] attempted to improve the performance of emotion recognition by: (i) utilizing the one-versus-all approach using SVM as the base classifier, and (ii) employing combinations of several preprocessing methods. The main finding was the significant role of the adopted preprocessing techniques on the predictive capability of the emotion detection system. Note that the classification problem is a multi-label problem. In the same way, authors in [10] focused on using preprocessing methods and an ensemble approach to enhance the classification performance. More precisely, Synthetic Minority Oversampling Technique (SMOTE) was applied to handle the imbalanced dataset problem, and three classifiers were utilized: SVM, Naive Bayes (NB), and K-Nearest Neighbors (KNN) algorithms, as well as an ensemble of the three considered classifiers. The evaluations were conducted using the SemEval-2018 dataset. The obtained results showed an improvement in the detected emotions in Arabic text when using the ensemble model.

Although utilizing traditional ML models has formed a strong baseline for emotion detection, understanding deep context is considered a critical issue. Consequently, recent research has concentrated on utilizing DL algorithms. For instance, authors in [11] employed deep Convolutional Neural Networks (CNNs) and compared them with three traditional ML classifiers (NB, SVM, and Multilayer Perceptron (MLP). The experimental findings reported the superiority of CNNs for detecting emotions. Another attempt performed by authors in [12] implemented Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks to create a targeted emotion detection model. These models were fed with three different text representation methods: (i) human-engineered features, (ii) deep-embedding features, and (iii) hybrid features. The results highlighted the superiority of the hybrid features for the three considered datasets: SemEval-2018 [1], Iraqi Arabic

Emotion Dataset (IAEDS) [13], and Arabic Emotion Tone Detection (AETD) [2] dataset. Authors in [14], utilized Bidirectional Long Short-Term Memory (BiLSTM) and AraVec, a pre-trained word embedding model for Arabic, to construct the intended Arabic emotion detection model. According to the reported findings, the model produced the most accurate predictions compared to SVM, RF, and fully connected neural networks. Moreover, some researchers reported that combining two DL models into a hybrid single model improved the overall effectiveness [15]. Despite many researchers demonstrating that DL methods outperform traditional ML because they learn semantic and syntactic features directly from text, they often require large labeled datasets, and their performance may degrade when trained on small and imbalanced Arabic textual datasets [16]. Recently, Pre-trained Language Models (PLMs) have been utilized for generating effective emotion recognition models. Recent research studies have reported that fine-tuned PLMs significantly enhance Arabic emotion recognition [17]. The reasons behind the effectiveness of these models are: (i) the underlying transformer-based architecture and (ii) the pre-training on huge corpora, which enables them to capture knowledge. Due to their noticeable impact in the field of NLP, ongoing research aims to improve their domain specificity and refine the fine-tuning methodologies. Models such as AraBERT [18], Multilingual Arabic BERT (MARBERT) [18], and QARiB [19] are specifically trained for Arabic textual data. It is interesting to mention that using PLMs is referred to as transfer learning, because the expertise gained during addressing one problem is employed to solve another problem that is different but related [19].

Few research studies have attempted to compare ML or DL with recent PLMs, such as the work conducted by authors in [20], in which SVM coupled with TF–IDF and Word2Vec was compared with BERT models (AraBERT [18] and ArabicBERT [21]) for the SemEval-2018 dataset. The observed outcomes showed that ArabicBERT significantly outperformed ML methods. However, this study only considered one ML algorithm, and therefore it cannot be considered comprehensive. Some studies attempted to combine different ML, DL, and PLMs together in a simple ensemble approach to benefit from all these approaches and improve the final resulting emotion recognition model [16].

Despite the progress, according to our recent research, there has been no comprehensive comparative study that systematically examines the different approaches employed to generate an effective Arabic emotion recognition model. The work presented in this paper addresses that gap. More precisely, the primary target of the research investigated in this paper is to construct an effective framework for emotion identification from Arabic textual data. To achieve this target, a fine-grained large-scale dataset is utilized. Moreover, classic ML, DL, PLMs, and advanced hybrid models are employed. As a result, a comprehensive experimental study that systematically evaluates the following four methodological approaches for fine-grained Arabic emotion detection is presented:

- Traditional ML classification models.
- DL classification models.
- Pre-trained Arabic language models.
- A hybrid classification model that combines the best-performing models using parallel and sequential strategies.

Consequently, the essential contributions of this research study can be summarized as outlined below:

- A systematic comparison of the main recognized approaches for fine-grained Arabic emotion detection.
- Practical recommendations for model selection based on performance.
- Offering reproducible experimental configurations to support researchers in the Arabic fine-grained emotion detection field.
- A novel hybrid model integrating the best-performing model from the main approaches.

II. THE ADOPTED METHODOLOGY

In this section, the experimental methodology for constructing the optimal emotion detection model is clarified. Figure 1 displays and summarizes the overall methodological framework. The entire methodological framework is based on the AETD [2] dataset, which is a benchmark dataset in the domain of Arabic NLP. The dataset consists of 10,065 text samples and seven human emotions categorized as: anger, fear, happiness, love, sadness, surprise, and sympathy. Each sample is assigned a single emotion. If no emotions are observed, a "none" emotion class is assigned. The proportion of classes is presented in Figure 2. The original study that introduced this dataset evaluated only two ML models, specifically NB and SVM, and reported NB as the best model. In this work, the evaluation is extended to include several conventional ML models, DL models, PLMs, and hybrid approaches. All experiments employ clearly defined parameters and k-fold cross-validation to ensure a fair comparison. Consequently, a comprehensive benchmark for future studies in Arabic emotion detection is provided.

The AETD dataset is cleaned and preprocessed, including tokenization, removal of stop words, punctuation, and special characters, as well as stemming, before it is used to train and test three main categories of models separately:

- The models constructed using conventional ML algorithms, in particular, Logistic Regression (LR), RC, SVM, NB, Decision Tree (DT), Extreme Gradient Boosting (XGBoost), and RF, are examined in this study.
- The models produced by DL algorithms, specifically Feedforward Neural Network (FNN), Bidirectional Gated Recurrent Unit (BiGRU), and BiLSTM, are investigated in this study.
- The models that were previously pre-trained on large Arabic corpora, including MARBERTv2 [17], Egyptian Arabic BERT (EgyBERTv2) [22], and Cross-Lingual Language Model (XLM-Roberta-base) [23], are considered in this study.

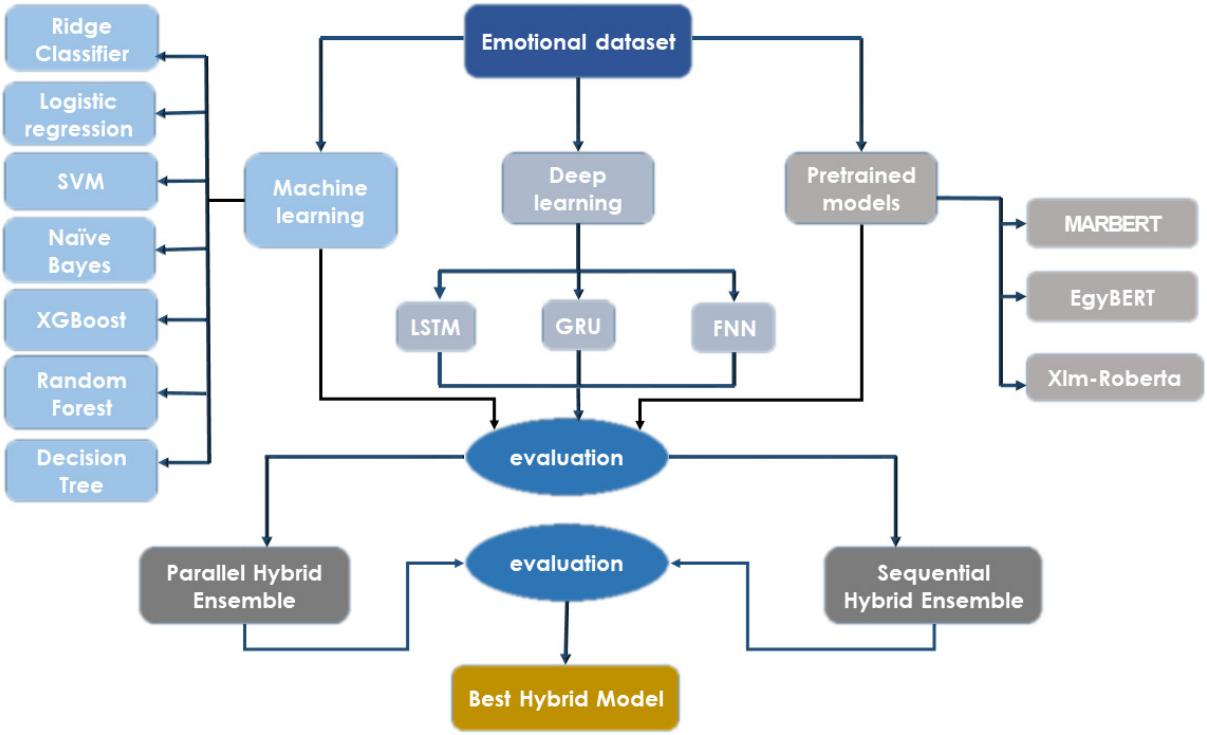

Fig. 1. The methodological framework.

Based on the evaluation, the best-performing model from each category is selected and employed as a base classifier in two main ensemble approaches: parallel and sequential. With respect to the parallel ensemble, the three base models are combined using a voting mechanism.

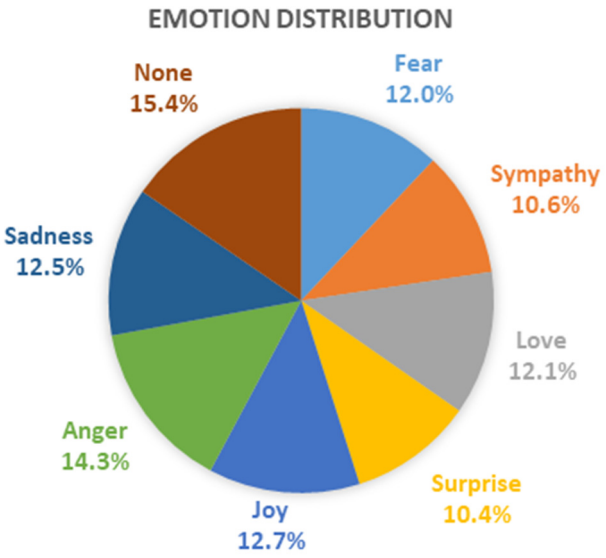

Fig. 2. The proportions of emotion classes in the AETD dataset.

Regarding the sequential ensemble, it can be viewed as a "boosting-inspired" model. After each training stage, the prediction correctness of each training sample is used to estimate its difficulty. Difficult samples are assigned larger weights to influence the training of the next model, either through loss weighting (for the LR stage) or weighted resampling (for the BiGRU stage). Figure 3 illustrates the overall sequential model construction, whereas Algorithm 1 provides a detailed description of the weight calculation and sample selection.

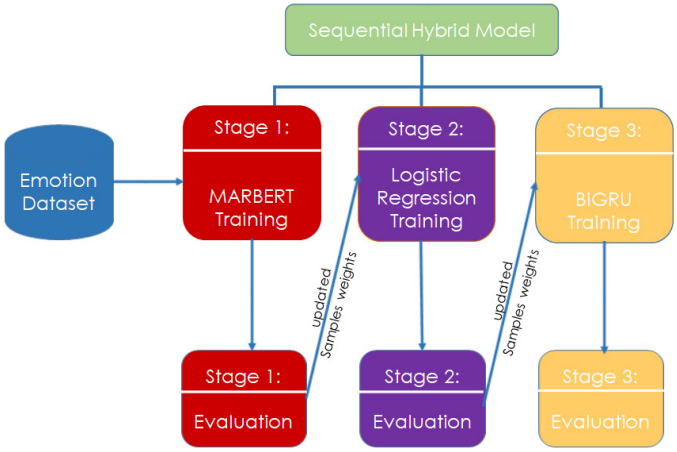

Fig. 3. The construction phase of the sequential hybrid model.

The algorithm constructs a sequential, "boosting-inspired" hybrid model. The dataset D and the number of folds K are the inputs to the algorithm. The construction process begins by splitting the dataset into K folds. For each cross-validation iteration, one subset is utilized for testing, and the rest folds are used for training. Each iteration starts with training MARBERTv2 and evaluating it using a validation set to generate class probability predictions for every sample. These probabilities are used to estimate samples' "hardness," where

samples receiving low probability for their correct label are considered harder to classify (line 9). The hardness scores are then transformed into weights (line 11).

More precisely, two parameters control the weighting process: (i) α, which controls the overall level of emphasis placed on difficult training samples and (ii) γ, which controls how strongly the emphasis increases as samples become more difficult. The resulting weights are then normalized to maintain training stability. In the experiments reported in this paper, α and γ were set to 1.5 and 2, respectively.

The next stage is LR training, where the calculated weights from the previous stage are used as input for the current stage (line 13). After that, the model is evaluated and probabilities are generated to be used again for estimating the hardness and the weights of the samples. These updated weights are used to resample the instances to the next stage; hard instances are highly represented in the next stage (line 15).

In the final stage, the resampled dataset is used to train the BiGRU model (line 18), which represents the final stage of the sequential architecture. The BiGRU is also evaluated by predicting probabilities of the samples in the evaluation set (line 19). The output of this algorithm is the sequential hybrid model, which consists of the three constructed models trained to focus on hard samples. With respect to model usage, the three base models are combined using average probabilities.

Algorithm 1: Sequential Hybrid Model Construction

1: Input: Dataset D, folds K
2: Output: The sequential hybrid model
3: Split D into K stratified folds
4: For each cross-validation iteration:
5: Designate one fold as the test, remaining folds as the training
6: Train MARBERTv2
7: Predict probabilities P_1
8: Compute the sample hardness:
9: $hardness_i = 1 - P_1(i, true\ class)$
10: Compute the sample weights:
11: $w_1(i) = 1 + \alpha(hardness_i)^{\gamma}$
12: Normalize the weights
13: Train LR using w_1 as sample weights
14: Predict probabilities P_2
15: Compute $hardness$ and weights w_2
16: Perform weighted resampling using w_2
17: Prepare the data for training the BiGRU model
18: Train BiGRU using the resampled data
19: Predict probabilities P_3 and evaluate BiGRU
20: End for

III. EXPERIMENTS AND RESULTS

This section analyzes and discusses the achieved results. Before discussing the produced results, it is interesting to note that an exploration of the dataset was conducted to gain insight into it using the word cloud visualization method. However, an overlap among emotion words was observed. Consequently, lexicon-based methods are not effective in this case. Four main result categories are reported in this section: (i) ML results, (ii) DL results, (iii) pre-trained models results, and (iv) hybrid models results. All reported results are obtained using a 5-fold cross-validation. For a broad assessment, every empirical result is reported using four well-known metrics, particularly accuracy, precision, recall, and F1-score.

A. Machine Learning Results

Table I presents an overview of the ML model results. A grid search mechanism was employed to find the most appropriate parameters for the different learning models used. Note that all models utilize the same text representation method, TF–IDF. In this study, a combination of word-level (1–2 grams) and character-level (3–6 grams) features was considered, with a maximum of 20 k features. The highest performance was achieved by the LR classifier.

TABLE I. ML RESULTS USING FOUR EVALUATION METRICS WITH THE BEST PARAMETER SETTINGS (UNMENTIONED PARAMETERS ARE SET TO DEFAULT)

Model	Acc.	Prec.	Rec.	F1	Parameters
LR	0.678	0.675	0.678	0.668	L2 regularization, C = 2.0, class-balanced
RC	0.667	0.663	0.667	0.660	L2 regularization (α = 1.0)
SVM	0.661	0.658	0.661	0.658	C = 2.0, squared hinge, linear kernel, class-balanced
XGBoost	0.638	0.636	0.638	0.622	400 trees; max depth = 6; learning rate = 0.08
NB	0.622	0.627	0.622	0.601	Smoothing α = 0.5
RF	0.605	0.604	0.605	0.573	400 trees, no maximum depth
DT	0.458	0.455	0.458	0.454	Gini criterion, no maximum depth

B. Deep Learning Results

As mentioned earlier, FNN, BiLSTM, and BiGRU were examined. To capture semantic and contextual information from the data, MARBERT was employed to generate embeddings for the considered DL models. The results are presented in Table II, which clearly demonstrates the superiority of BiGRU (F1-score = 0.742). Moreover, the best-performing DL model (BiGRU) demonstrates higher effectiveness than the most effective ML model (LR).

C. Pre-Trained Models Results

Regarding the pre-trained models, a grid search was conducted to identify the best hyperparameter configurations. The results for each model, along with their corresponding hyperparameter settings, are presented in Table III. From the table, it is clear that the MARBERTv2 model outperformed EgyBERTv2 and XLM-Roberta-base. This superiority is primarily due to the pre-training process: MARBERT was trained on larger and more diverse Arabic corpora compared to EgyBERTv2 and XLM-Roberta-base, which enables it to better understand different dialects, informal language, and patterns in Arabic.

TABLE II. DL RESULTS USING FOUR EVALUATION METRICS WITH THE BEST PARAMETER SETTINGS (UNMENTIONED PARAMETERS ARE SET TO DEFAULT)

Model	Acc.	Prec.	Rec.	F1	Hyperparameter setting
BiGRU	0.741	0.760	0.741	0.742	Opt = AdamW, HS = 256, NL = 2, DR = 0.4, BS = 16, EP = 8, LR = 1e-3, WD = 1e-4, MaxLen = 128
BiLSTM	0.740	0.749	0.737	0.737	Opt = AdamW, HS = 256, NL = 2, DR = 0.4, BS = 32, EP = 15, LR = 1e-3, WD = 1e-4, MaxLen = 128
FNN	0.722	0.734	0.717	0.716	Optimizer = AdamW, HS1 = 128, HS2 = 256, NL = 2, DR = 0.4, BS = 32, EP = 15, LR = 1e-3, WD = 1e-4

Opt = Optimizer, HS = Hidden Size, NL = Number of Layers, DR = Dropout Rate, BS = Batch Size, EP = Epochs, LR = Learning Rate, WD = Weight Decay.

TABLE III. PRE-TRAINED MODEL RESULTS USING FOUR EVALUATION METRICS WITH THE BEST PARAMETER SETTINGS

Model	Acc.	Prec.	Rec.	F1	Hyperparameter setting
MARBERTv2	0.795	0.795	0.798	0.794	Optimizer = AdamW, LR = 2e-5, WS = 0, WD = 0.01, EP = 4, BS = 16, MSL = 128, DR = default, Seed = 42, CW = No
EgyBERTv2	0.771	0.769	0.773	0.770	Optimizer = AdamW, LR = 2e-5, WS = 500, WD = 0.01, EP = 4, BS = 16, MSL = 128, DR = default, Seed = 42, CW = No
XLM-Roberta-base	0.706	0.702	0.705	0.705	Optimizer = AdamW, LR = 2e-5, WS = 100, WD = 0.01, EP = 5, BS = 16, MSL = 150, DR = 0.03, Seed = 42, CW = True

LR = Learning Rate, WS = Warmup Steps, WD = Weight Decay, EP = Epochs, BS = Batch Size, MSL = Maximum Sequence Length, DR = Dropout Rate, CW = Class Weights.

D. Parallel and Sequential Hybrid Results

In both the parallel and sequential hybrid models, MARBERTv2, BiGRU, and LR are integrated, as these were the best-performing models from the three examined categories.

For the parallel hybrid model, the base models are trained independently. Two main techniques are employed to combine the predictions: (i) hard voting, where each base model votes for a single class and all models have equal voting weight, and (ii) soft voting, where each base model generates a probability for each class, and the average probability across all models is calculated for each class. Table IV presents the obtained results for each combination technique. Soft voting significantly outperforms hard voting, indicating that using average probabilities instead of equal voting yields a higher-performing final hybrid model.

TABLE IV. PARALLEL HYBRID MODEL RESULTS USING FOUR EVALUATION METRICS

Model	Acc.	Prec.	Rec.	F1
Soft hybrid voting	0.790	0.796	0.788	0.790
Hard hybrid voting	0.787	0.794	0.784	0.787

For the sequential hybrid model, two pipelines are considered. Pipeline 1 begins with MARBERTv2, the best-performing model among MARBERTv2, BiGRU, and LR. Pipeline 2 ends with MARBERTv2, designed to investigate the impact of base model order on the final hybrid model's performance. The results, shown in Table V, indicate that the order of the base models significantly affects overall effectiveness. Specifically, Pipeline 1, which starts with MARBERTv2, achieved an F1-score of 0.802, whereas Pipeline 2, which does not start with MARBERTv2, yielded an F1-score of 0.788.

TABLE V. SEQUENTIAL HYBRID MODEL RESULTS USING FOUR EVALUATION METRICS

Model	Acc.	Prec.	Rec.	F1
Sequential model Pipeline 1	0.801	0.805	0.800	0.802
Sequential model Pipeline 2	0.787	0.795	0.784	0.788

IV. CONCLUSION

Research in Natural Language Processing (NLP) has highlighted that no single method or algorithm consistently outperforms others across all scenarios. The study presented in this paper aimed to serve as a benchmark for future research in emotion identification from Arabic textual data and to demonstrate the strengths and weaknesses of several available approaches.

Several approaches were explored in this research study, including: (i) Machine Learning (ML), (ii) Deep Learning (DL), (iii) Pre-trained Language Models (PLMs), and (iv) hybrid models. For the hybrid models, two main training strategies were adopted: (i) parallel and (ii) sequential. In the parallel hybrid models, the base models are independent, and their outputs are combined using either soft voting or hard voting. In the sequential hybrid model, each model depends on the previous model during training, and an investigation was conducted to evaluate the effect of the base model order on the performance of the final hybrid model.

The results indicate that the sequential hybrid model outperforms all single models and the parallel hybrid models, highlighting the effectiveness of combining models such that one model utilizes information from the training of the previous model, thus emphasizing learning on hard instances. Furthermore, the study revealed that the order of the base models within the sequential architecture significantly influences overall performance.

This study opens interesting directions for future research, including designing hybrid models that combine only the most effective pre-trained transformer models, as well as exploring more diverse Arabic datasets to improve generalization and robustness.

REFERENCES

[1] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "SemEval-2018 Task 1: Affect in Tweets," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, New Orleans, LA, USA, 2018, pp. 1–17, https://doi.org/10.18653/v1/S18-1001.

[2] A. Al-Khatib and S. R. El-Beltagy, "Emotional Tone Detection in Arabic Tweets," in *Computational Linguistics and Intelligent Text Processing: 18th International Conference, CICLing 2017*, Budapest, Hungary, 2017, pp. 105–114, https://doi.org/10.1007/978-3-319-77116-8_8.

[3] T. Al-Moslmi, M. Albared, A. Al-Shabi, N. Omar, and S. Abdullah, "Arabic senti-lexicon: Constructing publicly available language resources for Arabic sentiment analysis," *Journal of Information Science*, vol. 44, no. 3, pp. 345–362, June 2018, https://doi.org/10.1177/0165551516683908.

[4] A. Assiri, A. Emam, and H. Al-Dossari, "Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis," *Journal of Information Science*, vol. 44, no. 2, pp. 184–202, Apr. 2018, https://doi.org/10.1177/0165551516688143.

[5] J. Serrano-Guerrero, B. Alshouha, F. P. Romero, and J. A. Olivas, "Affective Knowledge-enhanced Emotion Detection in Arabic Language: A Comparative Study," *JUCS - Journal of Universal Computer Science*, vol. 28, no. 7, pp. 733–757, July 2022, https://doi.org/10.3897/jucs.72590.

[6] G. Badaro *et al.*, "EMA at SemEval-2018 Task 1: Emotion Mining for Arabic," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, New Orleans, LA, USA, 2018, pp. 236–244, https://doi.org/10.18653/v1/S18-1036.

[7] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256–265, Jan. 2017, https://doi.org/10.1016/j.procs.2017.10.117.

[8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, June 2017, https://doi.org/10.1162/tacl_a_00051.

[9] H. Mulki, C. Bechikh Ali, H. Haddad, and I. Babaoğlu, "Tw-StAR at SemEval-2018 Task 1: Preprocessing Impact on Multi-label Emotion Classification," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, New Orleans, LA, USA, 2018, pp. 167–171, https://doi.org/10.18653/v1/S18-1024.

[10] F. A. Q. Aljradi, M. Albared, A. S. Ghareb, and A. Thawaba, "An Ensemble Framework for Imbalanced Arabic Text-Based Emotion Analysis," *Journal of Science and Technology*, vol. 30, no. 1, pp. 60–71, 2025, https://doi.org/10.20428/jst.v30i1.2566.

[11] M. Baali and N. Ghneim, "Emotion analysis of Arabic tweets using deep learning approach," *Journal of Big Data*, vol. 6, no. 1, Oct. 2019, Art. no. 89, https://doi.org/10.1186/s40537-019-0252-x.

[12] N. Alswaidan and M. E. B. Menai, "Hybrid Feature Model for Emotion Recognition in Arabic Text," *IEEE Access*, vol. 8, pp. 37843–37854, 2020, https://doi.org/10.1109/ACCESS.2020.2975906.

[13] A. J. Almahdawi and W. J. Teahan, "A New Arabic Dataset for Emotion Recognition," in *Proceedings of the 2019 Computing Conference, Volume 2*, London, UK, 2019, pp. 200–216, https://doi.org/10.1007/978-3-030-22868-2_16.

[14] E. A. H. Khalil, E. M. F. E. Houby, and H. K. Mohamed, "Deep learning for emotion analysis in Arabic tweets," *Journal of Big Data*, vol. 8, no. 1, Oct. 2021, Art. no. 136, https://doi.org/10.1186/s40537-021-00523-w.

[15] A. Tolep, S. Mamikov, Z. Yakhiya, and A. Abdullayeva, "Sentiment Analysis of Social Network Data Using Traditional and Hybrid Deep Learning Models," *Engineering, Technology & Applied Science Research*, vol. 15, no. 6, pp. 28481–28488, Dec. 2025, https://doi.org/10.48084/etasr.12865.

[16] M. J. Althobaiti, "Arabic Emotion Recognition in Low-Resource Settings: A Novel Diverse Model Stacking Ensemble with Self-Training," *Applied Sciences*, vol. 13, no. 23, Nov. 2023, Art. no. 12772, https://doi.org/10.3390/app132312772.

[17] A. El-Sayed, M. Abougabal, and S. Lazem, "Benchmarking a large Twitter dataset for Arabic emotion analysis," *SN Applied Sciences*, vol. 5, no. 8, July 2023, Art. no. 224, https://doi.org/10.1007/s42452-023-05437-1.

[18] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online, 2021, pp. 7088–7105, https://doi.org/10.18653/v1/2021.acl-long.551.

[19] A. Abdelali, S. Hassan, H. Mubarak, K. Darwish, and Y. Samih, "Pre-Training BERT on Arabic Tweets: Practical Considerations." arXiv, Feb. 21, 2021, https://doi.org/10.48550/arXiv.2102.10684.

[20] N. Al-Twairesh, "The Evolution of Language Models Applied to Emotion Analysis of Arabic Tweets," *Information*, vol. 12, no. 2, Feb. 2021, Art. no. 84, https://doi.org/10.3390/info12020084.

[21] A. Safaya, M. Abdullatif, and D. Yuret, "KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Barcelona, Spain (Online), 2020, pp. 2054–2059, https://doi.org/10.18653/v1/2020.semeval-1.271.

[22] F. Qarah, "EgyBERT: A Large Language Model Pretrained on Egyptian Dialect Corpora." arXiv, Aug. 07, 2024, https://doi.org/10.48550/arXiv.2408.03524.

[23] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 8440–8451, https://doi.org/10.18653/v1/2020.acl-main.747.