Scientific Journal of King Faisal University:
Basic and Applied Sciences

Using Machine Learning to Analyze Emotions in Arabic and Dialectical Texts

Dina Abdelnaser Hamed [1], Ben Bella Said Tawfik [2] and Mohamed Abdullah Makhlouf [2]

[1] Department of Information Technology, Faculty of Information Technology and Computer Science, Sinai University, Arish, Egypt.
[2] Department of information systems, Faculty of computers and informatics and computer science, Suez Canal university, Ismailia, Egypt.

LINK	RECEIVED	ACCEPTED	PUBLISHED ONLINE	ASSIGNED TO AN ISSUE
https://doi.org/10.37575/b/sci/240014	21/03/2024	01/10/2024	01/10/2024	01/12/2024
NO. OF WORDS	**NO. OF PAGES**	**YEAR**	**VOLUME**	**ISSUE**
7690	9	2024	25	2

Abstract

Social media is an imperative necessity in contemporary life. People can easily express their emotions and share moments on social media by writing a few words. Organizations approach Twitter as a rich data source that may be used to study emotions, but while many efforts have focused on sentiment analysis from text, emotion classification has received less attention. Emotion analysis usually provides a more in-depth assessment of the author's feelings, and in this research, we propose a dialectal Arabic text emotion classification architecture that accurately classifies the expressions into four emotions (anger, joy, fear, and sadness). Considering the improvements in natural language processing (NLP), we investigated the Bidirectional encoder representations from transformers (BERT) model. We implemented our proposed ensemble model via a majority voting technique that merges the best three versions of the pre-trained BERT models that are considered state-of-the-art in the classification field. We compared the results of our model with eight other machine learning classifiers and ten versions of the BERT model. The proposed ensemble approach accomplished around 84%, however the highest accuracy of the other investigated models was 76%. The presented experiments were examined on the Arabic tweets' dataset for the EI-OC task provided by SemiEval, which contains 5600 tweets.

KEYWORDS
text classification, fine-tuning, voting technique, naive bayes, augmentation, transformers

CITATION
Hamed, D.A., Tawfik, B.B.S. and Makhlouf, M.A. (2024). Using machine learning to analyze emotions in Arabic and dialectical texts. *Scientific Journal of King Faisal University: Basic and Applied Sciences*, **25**(2), 22–30. DOI: 10.37575/b/sci/240014

1. Introduction

In our daily life, we use a variety of facial expressions, vocal activities, and written material to express our emotions, prompting researchers to conduct emotion analysis. The majority of recently presented studies concentrate on sentiment analysis as positive and negative (Medhat *et al.*, 2014), but few go deeper to extract emotions from written text. In this work, we focus on emotion analysis from written text. A large number of researchers have introduced various new techniques and methods for emotion analysis in several languages such as English, Chinese and Persian, but very few have looked at the Arabic language and dialects.

Arabic ranks as the fifth most spoken language in the world and is the official language of more than 27 countries. More than 456 million Arabs speak the language, and Arabic Social Networks are the fastest-growing social networking sites in terms of languages used. Over the last five years, more than 237 million Arabs have used social media (Istizada, 2023). Emotion significantly impacts human cognitive processes, such as perception, focus, memory retention, logical thinking, and problem-solving. It has a significant effect on attention, while stimulating behavior and activity. Emotion analysis is an important field of study faces many challenges because of how complicated natural languages are, and because it is difficult to comprehend and measure how people express their feelings mathematically. Studies are in continuous development to prove that emotion analysis is an extremely useful and powerful marketing tool that helps product managers understand customers' emotions and opinions. It is important to improve a promotion's success, product acceptance, and customer satisfaction.

Organizations are also trying to analyze posts, comments, and discussions to extract all possible information about whether or not they are interested in a particular topic (Storey & O'Leary, 2024) and the level of user satisfaction toward a specific product or service. They focused on emotional marketing by trying to stimulate people's emotions to buy products or services. On the other hand, emotion analysis can indicate the success or failure of some governmental or non-governmental institutions in some of the tasks assigned to them, such as safety, stability, and good performance. It can also be used as a special code between members of the same organization as a kind of privacy and to facilitate the workflow with high efficiency.

The novelty of this paper lies in proposing a robust adaptive model to accurately classify Arabic text into four emotions: anger, joy, fear, and sadness. Taking advantage of advances in natural language processing (Wolf *et al.*, 2020), we explored the BERT model and implemented a group approach that integrated the three best versions of the pre-trained BERT model via the majority voting technique. Our proposed approach competes with the state of the art classification.

We compared the performance of our approach to eight other machine learning classifiers and ten versions of the BERT model. The proposed classification method achieved an accuracy of about 84%, outperforming the highest accuracy of 76% achieved by other models. We conducted experiments on a dataset of Arabic tweets for the EI-OC mission provided by SemiEval, which included about 5,600 tweets. Our findings demonstrate the effectiveness of the proposed group approach in accurately categorizing emotions in the Arabic script.

The remainder of this work is organized as follows: Relevant works are shown in Section 2, background information is shown in Section 3, methodology and our model are presented in Section 4, experiments and results analysis are presented in Section 5, and the conclusion is presented in Section 6.

2. Related Work

In recent years, research connected to emotion analysis has made efforts to employ machine learning approaches. This section covers a number of studies that employ machine learning algorithms to analyze user opinions, sentiments, and product evaluations found in web material.

A. Singh *et al.* (2016) compared algorithms for supervised machine

Corresponding Author: Dina Abdelnaser Hamed 00201015668988, dina.hamed@su.edu.eg

learning classifications that were designed to classify data based on prior knowledge; analyze the accuracy, learning speed, complexity, and risk of overfitting; measure the effectiveness of supervised machine learning algorithms; and compare them broadly with several machine learning techniques. According to Abdullah and Shaikh (2018), the UNCC technology examines tweets in English and Arabic to determine emotions. They presented the identical design in both languages for each of the five challenges. The system's main input is a set of psycholinguistic characteristics and a mix of word2vec and doc2vec embeddings (for example, Affective Tweets from the Weka-package). They acquire much higher Spearman correlation scores when they use a fully connected neural network design.

Daood *et al.* (2017) focused on Arabic emotion categorization. They collected a set of Arabic tweets from the Levant and marked them with their desired sentiments and employed a number of ways to categorize user text messages in order to ascertain their emotional states. They analyzed the results of multiple machine learning algorithms, taking numerous factors into consideration to reach the best emotion recognition result. According to Mohammad *et al.* (2018), they are the results of the SemEval-2018 Task, which comprises numerous subtasks on extrapolating a person's level of productivity from their tweets. They generated tagged data from tweets in English, Arabic, and Spanish for each assignment. The separate tasks were Emotion Classification, Emotion Intensity Ordinal Classification, Emotion Intensity Regression, and Valence (Sentiment) Regression. They presented an overview of the techniques, resources, and tools employed by the participating organizations, focusing on the most beneficial. They also looked at the underpinnings of a trustworthy proclivity for a certain race or gender.

Context-aware Geted Recurrent Units(C-GRU) were introduced by A. E. Samy *et al.* (2018), which assessed the context (themes) of tweets and utilized them as an extra layer to infer the thoughts that the tweet was attempting to communicate. By determining the weights that each sub-modal contributed to the predictions (from subjects and sentences), the multi-modal model combined the two learned outputs.

Abdullah *et al.* (2020) developed an approach for recognizing and categorizing emotions in Arabic tweets by employing anger, sadness, pleasure, and disgust. The results of the studies demonstrated the usefulness of the suggested models, which enhanced the state of the art for categorizing Arabic tweets using support vector machines (SVM) and naive Bayes (NB), which produced the best results.

Alswaidan and Menai (2020) developed three models for emotion identification in Arabic text: a deep feature-based (DF) model, a human-engineered feature-based (HEF) model, and a combination model (HEF+DF). Comparing the performances of the suggested models on each emotion label allowed them to determine how well they performed on the IAEDS, AETD, and SemEval-2018 datasets. He also compared the model's results to those of other cutting-edge models. The results show that the HEF+DF model outperformed the DF and HEF models across all datasets.

Elnagar *et al.* (2020) thoroughly analyzed several deep learning models for categorizing Arabic text to see how well they performed and investigated the influence of using word2vec embedding models to improve classification task performance.

Khalil *et al.* (2021) created a Bi-LSTM deep learning model to categorize emotions (EC) in shared Arabic tweets for the SemEval-2018 competition. They have consolidated the dataset files into a single file to be used in the cross-validation technique. Aravec with CBOW was used in the word embedding stage. According to their statistics, Jaccard Accuracy was 0.498, Precision was 0.695, Recall was 0.551, and F1 Score was 0.615.

Kamila *et al.* (2022) used a multi-task framework to create four emotion labels (anger, joy, fear, and sorrow) with intensity values and three temporal orientation labels (past, present, and future) from user tweets. The detected tweets for each user were pooled to determine the user's temporal orientation and sentiment. They explored how users' emotional states and temporal orientations interact with one another. According to this research, anger and happiness are associated with future orientation, but sorrow and fear are associated with historical orientation.

Alzanin *et al.* (2022) recently assessed three classifiers, Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), and Random Forest (RF), to classify Arabic text tweets into five groups based on linguistic features and content. They also looked at two alternative textual representations: term frequency-inverse document frequency and word embedding with Word2vec. In our review of related works, we highlighted several studies that addressed emotion classification in Arabic text using various machine learning algorithms. While these studies provided valuable insights into different methodologies and techniques, they often lacked comprehensive evaluations that directly compare the effectiveness and performance metrics of the proposed approaches. Singh *et al.* (2016), for instance, compared supervised machine learning classifiers across accuracy, learning speed, and complexity but did not benchmark against a unified dataset or standard evaluation metrics. Similarly, Abdullah and Shaikh (2018) and Daood *et al.* (2017) focused on emotion detection in Arabic tweets using specific models like SVM and neural networks, yet their evaluations primarily emphasized individual model performance rather than comparative analysis against other state-of-the-art methods. Our work addresses this gap by systematically evaluating our ensemble pretrained BERT model against a diverse set of baseline classifiers, demonstrating significant improvements in accuracy and robustness across emotion categories, and provides a clearer perspective on the advancements and impact of our proposed methodology in the field of Arabic emotion classification.

3. Background

In this section, we will present a brief review of the transformer architecture and the main classification algorithms.

3.1. Transformer Architecture:

The transformer architecture is a deep learning model designed for natural language processing (NLP) introduced in 2017 that utilizes self-attention to capture long-range dependencies within input sequences (Vaswani *et al.* 2017). The transformer architecture has transformed the field of NLP by enabling models to attend to different parts of the input sequence and generate more accurate outputs. The transformer architecture consists of an encoder-decoder framework, with the encoder generating hidden representations of the input sequence and the decoder generating the output sequence. The self-attention mechanism computes attention weights for each input token, determining the level of attention each token should receive when computing the next hidden representation. This architecture has been widely used in NLP tasks, and pre-trained transformer models such as BERT and GPT-2 (Qian *et al.* 2022) have achieved state-of-the-art performance on many benchmarks (Yagi *et al.* 2023).

As seen in Figure 1, computers cannot understand words, thus in the word embedding stage, they take the input text and turn the words into vectors of numbers. The positional vector, which provides context based on the location of each word in the sentence, is then added to provide word embedding with context information in a vector of numbers. The transformer architecture consists of the encoder, decoder, and final linear layer. The linear layer takes the decoder's output as input and outputs it.

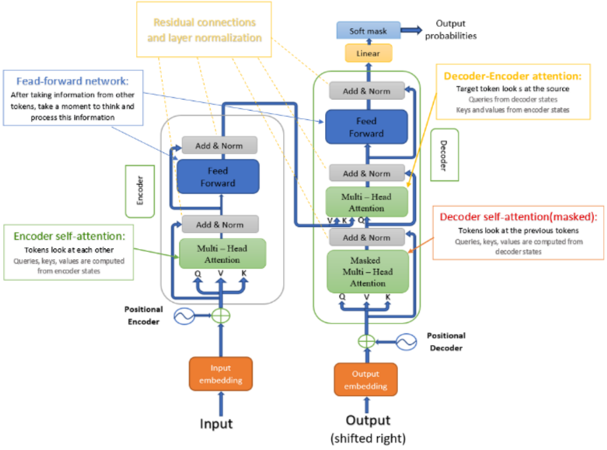
3.1.1. Encoder Layer

Each multi-head attention block (with padding mask) at the encoder layer receives three inputs: Q as question, V as value, and K as key. These are processed through linear (dense) layers before the multi-head attention function, which entails determining which portion of the input to focus on by creating distinct attention vectors and calculating the weighted average attention vector for each word in the phrase. The feed-for-word step is a basic feed-for-word neural network that is applied to each attention vector to make it more digestible so that it may be put to the next decoder block or linear layer based on its position.

3.1.2. Decoder Layer

The decoder module takes the output of the encoder as input and generates the output sequence, the masked multi-head attention (input should be masked) at the decoder layer displays as a vector of weighted values to show how strongly each word is connected to the other word in the same phrase. There is also multi-head attention (with padding mask) because Q receives the decoder's first attention block output, V and K receive the encoder output as inputs, and the attention weights describe the priority given to the decoder's input depending on the encoder's output. In other words, the decoder anticipates the next token by seeing the encoder output and self-attending to it. Point-wise feed-forward networks provide the same function as the encoder layer's equivalent step.

Figure 1. Transformers Architectures


Residual connections and layer normalization
Soft mask
Output probabilities
Linear
Fead-forward network:
After taking information from other tokens, take a moment to think and process this information
Decoder-Encoder attention:
Target token look s at the source
Queries from decoder states
Keys and values from encoder states
Add & Norm
Feed Forward
Decoder
Encoder
Add & Norm
Feed Forward
Add & Norm
Multi – Head Attention
V K Q
Add & Norm
Decoder self-attention(masked):
Tokens look at the previous tokens
Queries, keys, values are computed from decoder states
Encoder self-attention:
Tokens look at each other
Queries, keys, values are computed from encoder states
Add & Norm
Multi – Head Attention
Q V K
Masked Multi – Head Attention
Q V K
Positional Encoder
Positional Decoder
Input embedding
Output embedding
Input
Output (shifted right)

3.2. The Main Classification Algorithms:

3.2.1. Support Vector Machine Algorithm (SVM)

The Support Vector Machine is one of the most extensively used supervised learning algorithms for classification and regression problems (Ye *et al.* 2009). The goal of the SVM method is to find the optimal line or decision boundary that divides n-dimensional space into classes, allowing us to simply categorise new data points in the future. This ideal decision boundary is referred to as a hyperplane. In SVM, we employ two alternative classifiers (Support Vector classifier (SVC) and Linear Support Vector classifier (Linear SVC). The main distinction is that Linear SVC minimises squared hinge loss, whereas SVC minimises regular hinge loss. Linear SVC allows you to manually draw a 'hinge' string for loss parameters.

3.2.2. Naive Bayes Algorithm (NB)

NB is a classification approach that is based on the Bayes Theorem and the concept of predictor independence (Venkatesh *et al.* 2020). To put it simply, a Naive Bayes classifier believes that the existence of one feature in a class has nothing to do with the presence of any other feature. It organizes critical information and anticipates the target or dependent variable values generated by an independent or predictor variable. In NB, we employ two distinct algorithms (Multinomial NB and Bernoulli NB). Multinomial NB is concerned with counts for several features that occur, whereas Bernoulli NB is concerned with counts for a single feature that occurs as well as counts for the same feature that does not occur.

3.2.3. Stochastic Gradient Descent Algorithm (SGD)

SGD is a prominent and widely used method in Machine Learning algorithms (Yousaf *et al.* 2020). Gradient simply refers to the slope or tilt of a surface. To get to the lowest point on the surface, one must descend a slope.

3.2.4. Decision Tree Algorithm

The Decision Tree (DT) function generates a classification structure that resembles a tree (Mendonça *et al.* 2007). It divides the main facts into classes and predicts the values of the target or dependent variable that is constructed using an independent or predictor variable.

3.2.5. Random Forest Algorithm

Random Forest is a common data science technique for obtaining judgments based on random trees (Breiman, 2001). It is made up of several separate decision trees that work together as a single unit. The successful prediction model is the class that earns the most votes from all of the trees in the random forest.

3.2.6. K Neighbors Algorithm

K Neighbors Algorithm is a machine learning algorithm that is built on the Euclidean distance between instance (Kadhim, 2019). It predicts class labels for each instance and discovers the Nearest k examples by computing the smallest Euclidean distance between each instance and the others.

4. Methodology

In this stage, we will provide the dataset used in the research as well as our proposed approach for analyzing emotions from Arabic tweets using multiple machine learning classification algorithms and comparing the trials to determine the best methodology that can be applied in this sector. Using the Python programming language, we implement these models and assess the outcomes.

4.1. Data Set:

4.1.1. Original Dataset

In our research, we utilized a standard dataset of Arabic tweets given by SemiEval for the EI-OC task 1 [8]. This dataset contained 5600 tweets, 60% of the data, with 3,376 tweets for the training set divided into 889 for joy, 882 for anger, 877 for fear, and 728 for sadness. The testing and validation sets made up 40%, with 1,563 tweets for the test set divided into 448 for joy, 373 for anger, 372 for fear, and 370 for sadness, and 661 tweets for the development set divided into 224 for joy, 150 for anger, 146 for fear, and 141 for sadness. Each set contained the specified emotional labels: anger, fear, joy, and sadness.

4.1.2. Augmented Dataset

To expand the dataset, we used the contextual word embedding augmentation technique to take some sentences and replace words with other words with the same meaning as predicted by a label-conditioned bi-directional language model. For example, " مبحبش التهديد حتى لو في مصلحتي"was replaced with " مبحبش التهديد حتى لو عشان مصلحتي " and "شعرت باللامبالاة أثناء الاجتماع الطويل " was replaced with " شعرت بالملل أثناء الاجتماع الطويل", and then we appended the new sentence to the dataset. This step was only performed on the training set, which increased the number of tweets from 3,376 to 4,215.

4.2. Data preprocessing:

Text preprocessing is important in creating any word embedding model because it can greatly impact the outcomes. Given that our dataset was in Arabic, we performed a specific preprocessing to identify the most effective pattern. The primary preprocessing steps used are outlined in the following subsections:

4.2.1. Tokenization

The preprocessing module starts here. separates the textual input into words or tokens, each separated by a delimiter (such as a space or a punctuation mark). The tokenization procedure produces a list of terms (Grefenstette ,1999).

We used the Arabic-specific tokenization tools provided by libraries such as the Natural Language Toolkit (NLTK) or custom tokenization scripts tailored for our specific dataset. These tools ensure that the tokenization process preserves linguistic nuances and handles Arabic script complexities effectively. Each token typically represents a word or a sub-word and serves as the basic building block for further text processing tasks like sentiment analysis and emotion classification. For example:

Original Text: "أحب القراءة والكتابة في وقت الفراغ."

Word Tokenization Output: ['أحب', 'القراءة', 'والكتابة', 'في', 'وقت', 'الفراغ', '.']

Sub-word Tokenization Output: ['أحب', 'القراء', 'ة', 'والكتاب', 'ة', 'في', 'وقت', ' الفراغ", '.']

4.2.2. Normalization

It is necessary to preprocess the text to remove noise and normalize some letters to make Arabic text clearer and more helpful in enhancing classification, so we clean the data as follows:

a) Replace various characters, which can be written in a variety of ways, with the normal form as "أ","آ" replaced with "ا" .
b) Remove links and mentions.
c) Remove all diacritical marks as indicated in Table 1.
d) Remove the elongation 'tatweel' character '_', which is used to expand words (for example, the word 'خطيـــر' may convert to the normal form).
e) Remove all punctuation marks like periods, question marks, exclamation points, commas, colons, semicolons, dashes, hyphens, brackets, braces, parentheses, apostrophes, quotation marks, and ellipses.
f) Remove numbers.
g) Remove Latin characters (Aa, Zz).
h) Remove repeated characters. When describing an action, such as laughing" ههههههه ", users frequently purposefully repeat a character in a word: awe-inspiring" واااااااااو indignant, indignant"لاااااااااااااااا", etc. Every other repeating character was removed since we reasoned that a word could only have two repetitions.
i) We Replace emojis with their meaning in words, as shown in Table 2 (Singh *et al.* 2019), because emojis are common in social media and have great importance in enhancing the meaning of the sentence and expressing the writer's feelings, so we cannot ignore them.

Table 1. Diacritics

Diacritical marks التشكيل				
Tanween with Shaddah	Tanween تنوين	Short vowels with Shaddah شدة	Short vowels	Pronunciation
ـًّ	ـً	ـَّ	ـَ	fatHah فتحة
ٍّ	ـٍ	ـِّ	ـِ	Kasrah كسرة
ٌّ	ٌ	ـُّ	ـُ	DHammah ضمة
		ـّ	ـْ	Sukuun سكون

Table 2. Emojis and their textual description

Emoji	Description
😂	Face with tears of joy
😘	Face blowing a kiss
😁	Grinning face with smiling eyes
😌	Relieved face
😝	Squinting face with tongue
😥	Sad but relieved face
😡	Angry face
😭	Loudly crying face
😓	Downcast face with sweat
😰	Anxious face with sweat

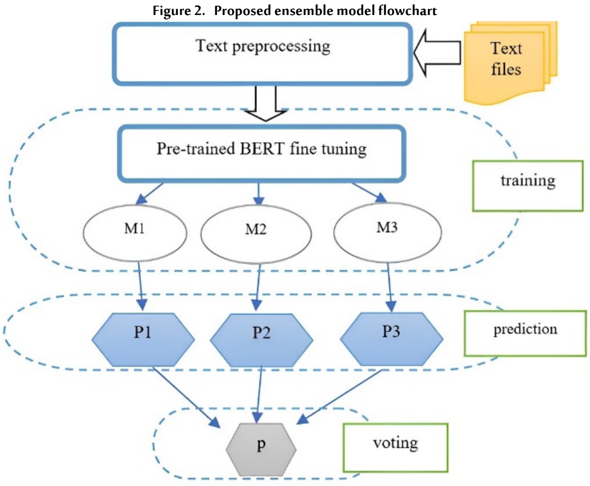
4.2.3. NLP Approach

In this approach, we use the transformer learning technique [19], where an NLP model trained on a very large dataset such as the Common Crawl and Wikipedia Corpus performs. Similar tasks on another dataset can be fine-tuned for specific tasks, which calls for a pre-trained model that adopts the mechanism of self-attention (Dai *et al.* 2020). It is like recurrent neural networks (RNNs) (Bullinaria, 2013) in processing sequential input data, but it works in parallel and inputs all at once, unlike other neural networks, so the attention mechanism provides context for every position in the input sequence.

4.2.4. Proposed Ensemble Model

BERT and GPT-2 are the most important transformer-based models. In this work, we will focus on a pre-trained BERT model and learn how to compare its various models and work on them to perform text classification. Figure 2 illustrates the overall architecture of our system, which consists of four steps: text preprocessing, pretrained BERT fine-tuning, prediction, and voting.

Figure 2. Proposed ensemble model flowchart



4.2.5. Pretrained BERT Fine-tuning

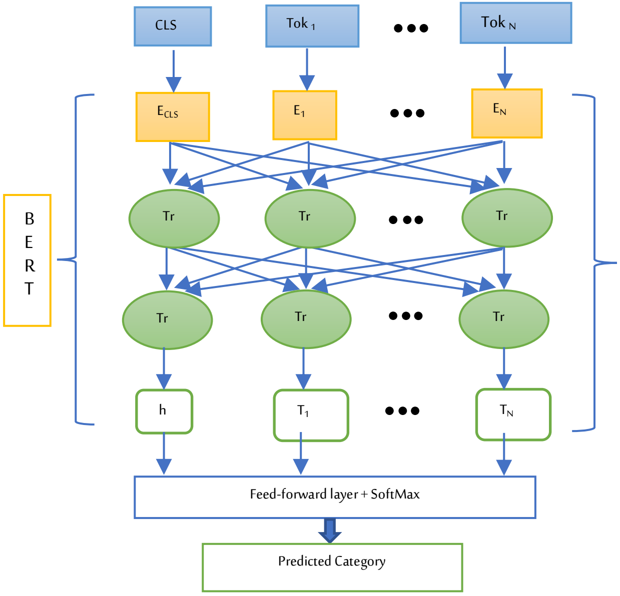
In this step, we work on the pre-trained BERT and fine-tune it for our Arabic data set. To do that, we connect the BERT outputs to an additional layer involving the SoftMax classifier after the fine-tuning to predict the text label. In the beginning, we tokenize every sentence into N tokens and add the [CLS] token at its beginning, the [CLS] token is a special token added to the beginning of each input sentence during tokenization. It stands for 'classification' and is utilized in tasks where the model needs to make a prediction based on the entire input sequence. The final hidden state corresponding to the [CLS] token after processing by BERT serves as the aggregated representation of the entire input sequence, which is then used as input to subsequent classification layers, such as the SoftMax classifier, to predict the text label.

After that, for each token i, we create an input representation called Ei by adding its vector embeddings. Then, we feed the Ei vectors into BERT and fine-tune its parameters using the corpus labelled data. As shown in Figure 3, We put h for the special [CLS] token's final hidden vector and Ti for the i-th input token's final hidden vector. Finally, to obtain the probability distribution for the projected output category, we use the final hidden state h as the representation of the entire text as an input for the feed-forward layer with the SoftMax classifier (Sun *et al.* 2019).

$$P(c/h) = Softmax\ (Mh) \qquad (1)$$

Equation (1) represents a way to calculate the probability of a target variable c, given a context or condition h. The equation uses a vector of scores Mh, which calculates the probability of each possible value of c given h. The Softmax function is applied to this vector to obtain a probability distribution over the possible values of c, where M is our task-specific parameter matrix. During fine-tuning, all parameters from BERT and M are trained jointly to maximize the log-probability of the correct category.

Figure 3. The fine-tuned BERT diagram

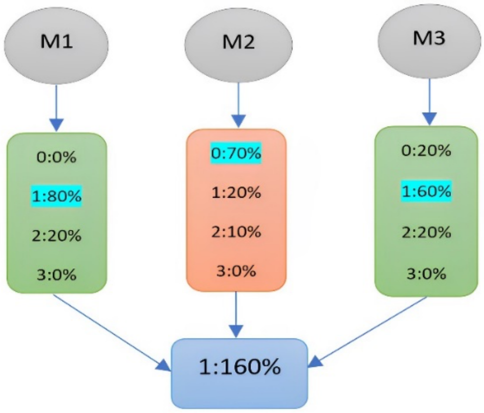


4.2.6. Voting Technique

We use a voting technique to reduce the complexity of ensemble models (Tiwari *et al.* 2024). In this step, each member of the ensemble model makes an equal contribution in parallel by giving a vote based on their predictions (Mohammed & Kora, 2022). There are two strategies of voting technique: soft voting by taking the average of the predicted results, or hard voting, that were adopted in our research, as shown in Figure 4. We take the best prediction by aggregating the scores of each predicted label for the three models and taking the highest score for the different results. We can see that if two or three make the same decision, we will take the top rating. If each model predicts a different answer, we will take the prediction that has the highest score.

If all opinions of all models have the same scores, then the prediction in this case will be the class that not only received the same score from all models, but also received the highest overall score across all models with the scores for each predicted label across all three models.

Figure 4. Hard voting technique

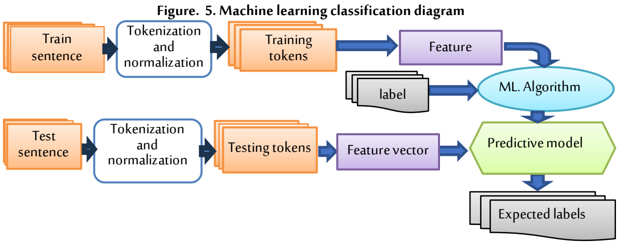


4.3. Machine Learning Approach:

In this approach, we applied eight supervised ML algorithms such as (linear SVC, SVC, multinomial NB, Bernoulli NB, SGD classifier, random forest classifier, decision tree classifier, and K Neighbors classifier. We show the ML architecture in a simple diagram in Figure 5 that shows the steps we followed for classifying the emotions of Arabic text into four emotion classes: anger, fear, joy, and sadness.

In the second phase, feature extraction and selection are carried out once the text has been prepared for processing. This section involves deleting irrelevant, redundant, and noisy data to obtain the best features that enhance the sentiment analysis procedure.

Figure. 5. Machine learning classification diagram



4.3.1. Features Selection

Many text features have been exploited in Sentiment Analysis (SA) research. The most popular ones are negation, n-grams, part of speech (POS), and term frequency. In our work, we use n-gram. The n-gram is a continuous string of n words taken from a prescribed sequence of text.

4.3.2. Features Extraction

In this phase, a piece of text is transformed into a feature with a specific weight to create each feature. There are various weighting methods, such as Boolean, Term Frequency (TF), and Term Frequency-Inverse Document Frequency (TF-IDF) (Rajaraman & Ullman, 2011). It uses the TF-IDF Vectorizer, a simple technique to vectorize text documents that transform sentences into a feature vector of numbers to be compatible with classifiers in the classification stage.

4.3.3. Machine Learning Classification

After getting, reading, preprocessing, and vectorizing the dataset by the TF-IDF Vectorizer, we proposed an adaptive emotion classification model for Arabic text into four emotional classes: anger, joy, fear, and sadness, using a broad, comprehensive set of machine learning classifiers. This model works on eight supervised ML algorithms such as linear SVC, SVC, multinomial NB, Bernoulli NB, decision tree classifier, SGD classifier, random forest classifier, and K-Neighbors classifier. Each algorithm is described in detail in this section.

5. Experiments and Results

In this paper, we work on several experiments to achieve the proposed model for Arabic text classification. We use the BERT model for Arabic text (Antoun *et al.* 2020), which has been pre-trained on 200 million phrases in Arabic, about 8.2 billion words, or 77 GB of text content. It features 512 maximum sequence lengths, 12 attention heads, 12 encoder blocks, 768 hidden dimensions, and 110 M parameters. We examine 10 new versions of this model and select the best three models to build our proposed ensemble model: bert-base-arabertv2, bert-medium-Arabic, and bert-large-arabertv02-twitter. We evaluate our proposed ensemble BERT model against other machine learning algorithms.

5.1. Analysis of Different BERT Models for Arabic Text Classification:

We trained ten different transformer models (Alammary, 2022) and evaluated their performance based on the training loss, validation loss, and F1-score. From these evaluations, we selected the three models that performed the best, which were the bert-base-arabertv2, bert-medium-Arabic, and bert-large-arabertv02-twitter models. Table 3 shows the results of the last epoch for each of the ten models, including their training loss, validation loss, and F1-score. The bert-base-arabertv2, bert-medium-Arabic, and bert-large-arabertv02-twitter models had the lowest training and validation losses and the highest F1-score, indicating that they performed the best overall. Therefore, we used these three models in the validation and test phases of our proposed ensemble model.

Table 3. Results of last epoch in each BERT model

Transformers model name	Training Loss	Validation Loss	F1-score
aubmindlab/bert-base-arabertv2	0.817400	0.972181	0.711044
aubmindlab/bert-large-arabertv2	1.403100	1.390501	0.220877
asafaya/bert-mini-Arabic	0.935900	1.096127	0.562784
asafaya/bert-medium-Arabic	0.710900	1.110692	0.692890
asafaya/bert-large-Arabic	1.400800	1.421856	0.226929
Distilbert-base-multilingual-cased	0.856100	1.223140	0.614221
nlptown/bert-base-multilingual-uncased-sentiment	0.816600	1.338099	0.594554
bert-large-arabertv02-twitter	0.664000	0.760911	0.757943
xlm-roberta -base	1.002700	1.036626	0.630862
xlm-roberta-large	1.392200	1.418853	0.220877

5.2. Evaluation Experiments:

As demonstrated in this experiment, we made a comparison between training the three selected models on the original data set and the augmented data set after appending the augmented sentences.

The results presented in Table 4 show the performance of the three selected BERT models of the last epoch in the training step, namely bert-base-arabertv2, bert-medium-Arabic, and bert-large-arabertv02-twitter, on both the original and augmented datasets. The models were evaluated based on their training loss, validation loss, and F1 score.

The bert-large-arabertv02-twitter model achieved the highest F1-score of 0.757943 on the original dataset and 0.753404 on the augmented dataset, followed by the bert-base-arabertv2 with an F1-score of 0.715582 on the original dataset and 0.670197 on the augmented dataset, and the bert-medium-Arabic with an F1-score of 0.694402 on the original dataset and 0.682300 on the augmented dataset.

Overall, the results suggest that the performance of the models varied depending on the quality of the dataset, with models performing better on the original dataset than the augmented dataset, which means that adding synonyms significantly confuses the results.

Table 4. Results of last epoch in the 3 best BERT models we used in our proposed model

Model	Faze	Training Loss	Validation Loss	F1-Score
bert-base-arabertv2	original	0.781600	0.896417	0.715582
	augmented	0.578800	1.106806	0.670197
bert-medium-Arabic	original	0.780700	0.910520	0.694402
	augmented	0.574400	1.196405	0.682300
bert-large-arabertv02-twitter	original	0.664000	0.760911	0.757943
	augmented	0.548300	0.914224	0.753404

We performed the voting step using the three best-performing models (bert-base-arabertv2, bert-medium-Arabic, and bert-large-arabertv02-twitter) and found that the F1-score increased by 9%. We then validated our ensemble model on a new dataset that included augmented sentences and compared the results with those obtained from the original dataset. However, we found that the results obtained using the augmented dataset were worse than those obtained using the original dataset, as shown in Table 5. The results suggest that the proposed ensemble model did not perform better on the augmented dataset, contrary to the researchers' expectations, due to the complexity of the Arabic language and the fact that a word carries more than one meaning depending on its presence in the sentence and its relationship to the words next to it. The augmentation process caused dispersion in the model used and led to worse results than using the original database.

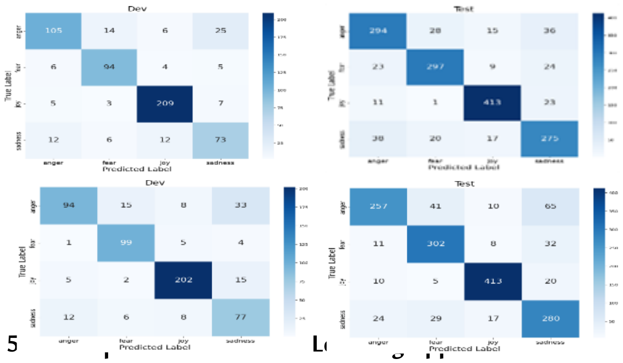
Table 5. Classification report

	Dev (validation set)-original			**Test-original**		
emotion	precision	recall	f1-score	precision	recall	f1-score
anger	0.82	0.7	0.76	0.8	0.79	0.8
fear	0.8	0.86	0.83	0.86	0.84	0.85
joy	0.9	0.93	0.92	0.91	0.92	0.92
sadness	0.66	0.71	0.69	0.77	0.79	0.78
accuracy			0.82			0.84
	Dev-augmented			**Test-augmented**		
emotion	precision	recall	f1-score	precision	recall	f1-score
anger	0.84	0.63	0.72	0.85	0.69	0.76
fear	0.81	0.91	0.86	0.80	0.86	0.83
joy	0.91	0.9	0.9	0.92	0.92	0.92
sadness	0.6	0.75	0.66	0.71	0.80	0.75
accuracy			**0.81**			**0.82**

In our research paper, we analyzed the results obtained from our emotion classification model and used a confusion matrix to clearly visualize the performance of the model on both the original dataset and the augmented dataset. Figure 6 (a) displays the confusion matrix for the validation results on the original dataset, representing the true prediction labels and the missed ones for each of the four classes. The diagonal line of the confusion matrix shows the number of predictions where the classifier correctly predicted the emotion label, while the other numbers indicate the missed predictions. Similarly, Figure 6 (b) displays the results of the test step on the original dataset.

To further evaluate the effectiveness of our model, we also analyzed the performance on the augmented dataset. Figure 6 (c) represents the validation results on the augmented dataset, and Figure 6 (d) displays the results of the test step on the augmented dataset. By comparing the results from the original and augmented datasets, we can observe a clear difference in the number of missed predictions between different emotions. Specifically, we found that the highest number of missed predictions was between anger and sadness. These findings suggest that our model may need further improvement to better distinguish between these two emotions. Overall, our analysis of the confusion matrix provides a comprehensive understanding of the performance of our emotion classification model and highlights areas for potential improvement.

Figure 6. Confusion matrices for dev. and test results on original and augmented datasets



5 L

Our research aimed to compare the performance of eight machine learning algorithms with three n-grams (uni-gram, bi-gram, tri-gram) in text classification (Wynne & Wint, 2019) (Euna *et al.* 2023). We evaluated the performance of each algorithm by calculating accuracy, precision, recall, and f1-score, and summarized the results in Table 6. Our findings indicate that the uni-gram approach provided the best results across all algorithms tested. In addition, we observed that the linear support vector classifier (SVC) achieved the highest accuracy, with a F1-score of 59.8%.

However, we also developed a proposed ensemble model and compared its performance with that of the eight machine learning algorithms tested. Our experimental results showed that our proposed ensemble model achieved a significant improvement in accuracy, with an F1-score of 84%, compared to the 60% accuracy obtained by the linear SVC model.

These findings demonstrate that our proposed ensemble model offers a promising approach to improving the accuracy and reliability of text classification tasks and has important implications for the field of natural language processing. By using a combination of several machine learning techniques, our ensemble model was able to outperform the individual models, highlighting the value of ensemble techniques in improving text classification performance. F1-score is the harmonic mean of a system's precision and recall values. It can be calculated by the following formula (Yacouby & Axman, 2020) to make a clear comparison with the same measure. By calculating the F1-score for the classifiers, the linear SVC classifier achieved the best result with 59.8%.

$$F1 - score \ = \ 2 * (precision * recall)/(precision + recall) \qquad (2)$$

Table 6. results of ML classifiers

Algorithm	gram	accuracy	precision	Recall
Linear SVC	1	0.601	0.598	0.601
	2	0.600	0.598	0.600
	3	0.596	0.595	0.596
SVC	1	0.560	0.612	0.560
	2	0.486	0.607	0.486
	3	0.440	0.630	0.440
MultinomialNB	1	0.581	0.628	0.581
	2	0.568	0.624	0.568
	3	0.561	0.622	0.561
Bernoulli-NB	1	0.511	0.615	0.511
	2	0.375	0.640	0.375
	3	0.316	0.682	0.316
SGD Classifier	1	0.585	0.580	0.585
	2	0.589	0.589	0.589
	3	0.587	0.585	0.587
Decision Tree	1	0.289	0.318	0.289
	2	0.289	0.318	0.289
	3	0.289	0.318	0.289
Random Forest	1	0.239	0.362	0.239
	2	0.237	0.278	0.237
	3	0.235	0.056	0.235
K-neighbors	1	0.432	0.517	0.432
	2	0.423	0.513	0.423
	3	0.419	0.502	0.419

5.4. Comparison With Previous Works:

Our research focused on evaluating the performance of our proposed ensemble BERT model compared to previous related work that used the same dataset. Our objective was to determine the effectiveness of our approach in tackling the SemEval 2018 EL-OC task1 dataset, a challenging task in the field. Our experimental results showed that our ensemble BERT model achieved the best results in accuracy, with an improvement of +8.5% compared to the previous best-performing model.

To compare the performance of our proposed model with previous work, we tested our results over the SemEval 2018 EL-OC task1 dataset and compared them with the results published in previous publications. Specifically, we compared our results with those obtained by Kamila *et al.* (2022). Our proposed model achieved an F1-score of 0.83 and an accuracy of 0.84, which represents a significant improvement over the results achieved by Kamila *et al.* (2022), who achieved an F1-score of 0.731 and an accuracy of 0.753.

Overall, our results demonstrate that our proposed ensemble BERT model is highly effective in addressing the SemEval 2018 EL-OC task1 dataset, outperforming the previous state-of-the-art models. These findings have important implications for the field of natural language processing, as they offer a promising approach for improving the accuracy and reliability of text classification tasks.

6. Conclusion

This work represented an ensemble pretrained BERT model that provides emotion classification for Arabic text in four classes (anger, joy, fear, and sadness) using a voting technique between the best three models in the Transformers library, which is a powerful, modern state-of-art model that has improved the accuracy of results from 76% to 84%. We merged these models and compared the results with other experiments on eight machine learning classifiers: linear SVC, SVC, multinomial NB, Bernoulli NB, SGD classifier, decision tree classifier, random forest classifier, and K-Neighbors classifier, implemented using three different n-grams with the TF-IDF feature values technique, using the Arabic tweets dataset given by the SemiEval competition for the El-oc task. Our method produced excellent performance with 82% validation accuracy and 84% testing accuracy.

In the future, we aim to apply the model we proposed to a larger dataset, work on the NLP technique to resolve natural language issues like the sarcasm detection issue and assess its impact on the results of emotion detection.

While this work has demonstrated potential approaches to emotion classification for Arabic text, the Arabic inflectional system, or the so-called Eraab, can be used to broaden the scope of this research, which has already shown some promising methods for sentiment categorization for Arabic text. We think that including the Eraab system in the framework for emotion analysis may enhance the results.

Biographies

Dina Abdelnaser Hamed

Department of Information Technology, Faculty of Information Technology and Computer Science, Sinai University, Arish, Egypt, Mobile +201015668988, Email dina.hamed@su.edu.eg

Dina is an Egyptian Teaching Assistant in the faculty of information technology and computer science at Sinai University. She has over ten years of experience in teaching and academic field and received her bachelor's degree in information technology at Sinai University, Egypt. She teaches programming, artificial intelligence, computer graphics, and logic subjects, and she is proficient in programming languages such as c++, c sharp, python, and mat lap. Her research interests include machine learning, data mining, and text classification.

ORCID: 0000-0003-1430-2010

Ben Bella Said Tawfik

Department of information systems, Faculty of computers and informatics and computer science, Suez Canal university, Ismailia, Egypt, Mobile +201223761595, Email benbellat@ci.suez.edu.eg

Prof. Ben Bella is an Egyptian professor in the faculty of computers and information at Suez Canal University with over 30 years of experience in the computer science field. He received his Ph. D. degree from the military technical college in 1986 and 1990 and from Colorado State University in 1998, respectively. He has published 20 ISI/Scopus-indexed articles with the largest global publishers including Elsevier, IJECE, Symmetry, IEEE. His research fields are related to computer networks, image processing, information systems, pattern recognition, and wireless sensor networks.

ORCID: 0000-0001-9352-7538.

Mohamed Abdullah Makhlouf

Department of information systems, Faculty of computers and informatics and computer science, Suez Canal university, Ismailia, Egypt, Mobile +201001263049, Email m.abdallah@ci.suez.edu.eg

Prof. Makhlouf is an Egyptian Professor in faculty of computers and information, Suez Canal university. He received his Ph. D. degree from Faculty of Science, Zagazig University. He got the post-doctoral studies in Computer science from Granada University Spain in 2016. He has published 14 ISI/Scopus-indexed articles in (IEEE Access, Symmetry, Journal of King Saud University, IJCSNS) His research interests: Machine learning, data mining, intelligent Bioinformatics, Decision support systems and predictive models.

ORCID: 0000-0002-8854-4912

References

Abdullah, M. and Shaikh, S. (2018). Teamuncc at SemEval-2018 Task 1: Emotion detection in English and Arabic tweets using deep learning. *In Proceedings of the 12th International Workshop on Semantic Evaluation*, **n/a**(n/a), 350–7. DOI**:** 10.18653/v1/S18-1

Abdullah, M., AlMasawa, M., Makki, I., Alsolmi, M. and Mahrous, S. (2020). Emotions extraction from Arabic tweets. *International Journal of Computers and Applications*, **42**(7), 661–75. DOI: 10.1080/1206212X.2018.1482395

Alammary, A.S. (2022). BERT models for Arabic text classification: A systematic review. *Applied Sciences*, **12**(11), 5720. DOI: 10.3390/app12115720

Alswaidan, N. and Menai, M.E.B. (2020). Hybrid feature model for emotion recognition in Arabic text. *IEEE Access*, **8**(n/a), 37843–54. DOI: 10.1109/ACCESS.2020.2975906

Alzanin, S.M., Azmi, A.M. and Aboalsamh, H.A. (2022). Short text classification for Arabic social media tweets. *Journal of King Saud University-Computer and Information Sciences*, **34**(9), 6595–604. DOI: 10.1016/j.jksuci.2022.03.020

Antoun, W., Baly, F. and Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. *ArXiv Preprint ArXiv:2003.00104*. **n/a**(n/a), 9–15. DOI: 10.48550/arXiv.2003.00104

Breiman, L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32. DOI: 10.1023/A:1010933404324

Bullinaria, J.A. (2013). Recurrent neural networks. *Neural Computation: Lecture*, **12**(n/a), 1–20.

Dai, B., Li, J. and Xu, R. (2020). Multiple positional self-attention network for text classification. *In Proceedings of the AAAI Conference on Artificial Intelligence*, **34**(5), 7610–7. DOI: 10.1609/aaai.v34i05.6261

Daood, A., Salman, I. and Ghneim, N. (2017). Comparison study of automatic classifiers performance in emotion recognition of Arabic social media users. *Journal of Theoretical and Applied Information Technology*, **95**(19), n/a.

Elnagar, A., Al-Debsi, R. and Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing and Management*, **57**(1), 102121. DOI: 10.1016/j.ipm.2019.102121

Euna, N.J., Hossain, S.M.M., Anwar, M.M. and Sarker, I.H. (2023). Content-based spam email detection using an N-gram machine learning approach. In: S. Nazmul , S.A. Mohammad , M Shamim , K. ASM (eds) *Applied Intelligence for Industry 4.0*. England, Oxon, Chapman and Hall.

Grefenstette, G. (1999). Tokenization. In: van Halteren, H. (eds) *Syntactic Wordclass Tagging. Text, Speech and Language Technology, vol 9*. Springer, Dordrecht. DOI: 10.1007/978-94-015-9273-4_9

Istizada (2023). *Complete List of Arabic Speaking Countries*. Available at: https://istizada.com/complete-list-of-arabic-speaking-countries/ (assessed on 15/8/2024)

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, **52**(1), 273–92. DOI: 10.1007/s10462-018-09677-1

Kamila, S., Hasanuzzaman, M., Ekbal, A. and Bhattacharyya, P. (2022). Investigating the impact of emotion on temporal orientation in a deep multitask setting. *Scientific Reports*, **12**(1), 493. DOI: 10.1038/s41598-021-04331-3

Khalil, E.A.H., Houby, E.M.E. and Mohamed, H.K. (2021). Deep learning for emotion analysis in Arabic tweets. *Journal of Big Data*, **8**(1), 136. DOI: 10.1186/s40537-021-00523-w

Medhat, W., Hassan, A. and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, **5**(4), 1093–113. DOI: 10.1016/j.asej.2014.04.011

Mendonça, L.F., Vieira, S.M. and Sousa, J.M.C. (2007). Decision tree search methods in fuzzy modeling and classification. *International Journal of Approximate Reasoning*, **44**(2), 106–23. DOI: 10.1016/j.ijar.2006.07.004

Mohammad, S., Bravo-Marquez, F., Salameh, M. and Kiritchenko, S. (2018). Semeval-2018 task 1: Affect in tweets. In: *Proceedings of the 12th International Workshop on Semantic Evaluation*, **n/a**(n/a), 1–17. DOI: 10.18653/v1/S18-1001

Mohammed, A. and Kora, R. (2022). An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, **34**(10), 8825–37. DOI: 10.1016/j.jksuci.2021.11.001

Qian, T., Xie, A. and Bruckmann, C. (2022). Sensitivity analysis on transferred neural architectures of bert and gpt-2 for financial sentiment analysis. *arXiv preprint arXiv:2207.03037*. DOI: 10.48550/arXiv.2207.03037

Rajaraman, A. and Ullman, J.D. (2011). *Mining of massive datasets*. 2nd edition. Stanford University, California, USA: Cambridge University Press. DOI: 10.1017/CBO9781139924801

Samy, A.E., El-Beltagy, S.R. and Hassanien, E. (2018). A context integrated model for multi-label emotion detection. *Procedia Computer Science*, **142**(n/a), 61–71. DOI: 10.1016/j.procs.2018.10.461

Singh, A., Blanco, E. and Jin, W. (2019). Incorporating emoji descriptions improves tweet classification. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, **1** (n/a), 2096–101. DOI: 10.18653/v1/N19-1214

Singh, A., Thakur, N. and Sharma, A. (2016). A review of supervised machine learning algorithms. In**:** *3rd International Conference on Computing for Sustainable Global Development (INDIACom),* New Delhi, India, 16–18 /3/ 2016.

Storey, V.C. and O'Leary, D.E. (2024). Text analysis of evolving emotions and sentiments in COVID-19 Twitter communication. *Cognitive Computation*, **16**(4), 1834–57. DOI: 10.1007/s12559-022-10025-3

Sun, C., Qiu, X., Xu, Y. and Huang, X. (2019). How to fine-tune bert for text classification?. In: *Chinese Computational linguistics: 18th China National Conference, CCL 2019*, Kunming, China, 18-20/10/2019. DOI: 10.48550/arXiv.1905.05583

Tiwari, D., Nagpal, B., Bhati, B.S., Gupta, M., Suanpang, P., Butdisuwan, S. and Nanthaamornphong, A. (2024). SPSO-EFVM: A Particle Swarm Optimization-Based Ensemble Fusion Voting Model for Sentence-Level Sentiment Analysis. *IEEE Access*, **12**(n/a), 23707–24. DOI: 10.1109/ACCESS.2024.3363158.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N. and Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems. In: *31st Conference on Neural Information Processing Systems (NIPS 2017),* Long Beach, CA, USA. 04-09/12/2017.

Venkatesh, R., K.V., Ranjitha and Venkatesh Prasad, B.S. (2020). Optimization scheme for text classification using machine learning Naïve Bayes classifier. In: *ICDSMLA 2019: Proceedings of the 1st*

International Conference on Data Science, Machine Learning and Applications, **n/a**(n/a), 576–86. DOI: 10.1007/978-981-15-1420-3_61

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A. and Rush, A.M. (2020). Transformers: State-of-the-art natural language processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, **n/a**(n/a), 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6

Wynne, H.E. and Wint, Z.Z. (2019). Content-based fake news detection using n-gram models. In: *Proceedings of the 21st International Conference on Information Integration and Web-based Applications and Services*, **n/a**(n/a), 669–73. DOI: 10.1145/3366030.3366116

Yacouby, R. and Axman, D. (2020). Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models. *In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, **n/a**(n/a), 79–91. DOI: 10.18653/v1/2020.eval4nlp-1.9

Yagi, S., Elnagar, A. and Fareh, S. (2023). A benchmark for evaluating Arabic word embedding models. *Natural Language Engineering*, **29**(4), 978–1003. DOI: 10.1017/S1351324922000444

Ye, Q., Zhang, Z. and Law, R. (2009). Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, **36**(3), 6527–35. DOI: 10.1016/j.eswa.2008.07.035

Yousaf, A., Umer, M., Sadiq, S., Ullah, S., Mirjalili, S., Rupapara, V. and Nappi, M. (2020). Emotion recognition by textual tweets classification using voting classifier (LR-SGD). *IEEE Access*, **9**(n/a), 6286–95. DOI: 10.1109/ACCESS.2020.3047831