computers

MDPI

Article

Classification of Arabic Poetry Emotions Using Deep Learning

Sakib Shahriar, Noora Al Roken and Imran Zualkernan *

Department of Computer Science and Engineering, American University of Sharjah, Sharjah 26666, United Arab Emirates
* Correspondence: izualkernan@aus.edu

Abstract: The automatic classification of poems into various categories, such as by author or era, is an interesting problem. However, most current work categorizing Arabic poems into eras or emotions has utilized traditional feature engineering and machine learning approaches. This paper explores deep learning methods to classify Arabic poems into emotional categories. A new labeled poem emotion dataset was developed, containing 9452 poems with emotional labels of joy, sadness, and love. Various deep learning models were trained on this dataset. The results show that traditional deep learning models, such as one-dimensional Convolutional Neural Networks (1DCNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) networks, performed with F1-scores of 0.62, 0.62, and 0.53, respectively. However, the AraBERT model, an Arabic version of the Bidirectional Encoder Representations from Transformers (BERT), performed best, obtaining an accuracy of 76.5% and an F1-score of 0.77. This model outperformed the previous state-of-the-art in this domain.

Keywords: text classification; deep learning; Arabic language; poetry; natural language processing; emotion

1. Introduction

Emotion is a fundamental element in communication because it helps us understand the conveyed message more comprehensively. We recognize the importance of emotion when using other communication methods, such as emails or text messages. While emotions are essential to human-to-human interaction, researchers have extended this contact to machines. Emotion recognition can be applied in numerous applications, such as extracting the public emotional reactions towards political, economic, or company decisions [1], improving the fields of human–computer interaction, psychology, e-learning, robotics, virtual reality (VR), and business [2], as well as analyzing opinions, current events, and human interests [3]. Manual emotion detection is time-consuming; therefore, machine learning and deep learning algorithms can be utilized to perform automatic emotion detection from texts.

Most existing work in emotion classification is applied in English, with only a few examples in Arabic. Almost 420 million speakers speak Arabic, divided into three forms: classical Arabic, modern standard Arabic (MSA), and Colloquial Arabic. Classical Arabic is found in the Quran, while MSA is a modern form of Arabic used by the media. Colloquial Arabic represents the different Arabic dialects spoken in various regions. Research on Arabic poetry is limited due to the structural complexity of the text, which requires mastery of the rules of the Arabic language. Detecting emotions in a poem's text is challenging as it requires the context of the entire text. A simple approach, such as looking for keywords within a poem, will often not provide successful classification. Moreover, poems displaying overlapping emotions, such as love and sadness, can be tricky even for humans. This research classifies emotions in Arabic poems using deep learning algorithms. To train the deep learning models, we introduce a novel Arabic poetry emotion dataset containing over nine thousand poems belonging to three emotion categories, namely joy, sadness, and love. As opposed to most existing work in Arabic text classification, which is machine learning-based, this work uses deep learning algorithms. To the best of our knowledge,

Citation: Shahriar, S.; Al Roken, N.; Zualkernan, I. Classification of Arabic Poetry Emotions Using Deep Learning. *Computers* **2023**, *12*, 89. https://doi.org/10.3390/computers12050089

Received: 10 March 2023
Revised: 10 April 2023
Accepted: 18 April 2023
Published: 22 April 2023

this is the first deep learning approach for detecting emotions in Arabic poems. The key contributions of this work are:

- It introduces an Arabic poem dataset with labeled emotions.
- It explores various deep learning models and transformer-based models to classify Arabic poems by emotions.
- It provides a performance comparison between deep learning and transformer-based models for Arabic poem classification.

The rest of the paper is organized as follows. Section two introduces the related works and describes deep learning algorithms. Section three describes the methodology, which includes the data collection and classification methods. The results are discussed in section four. Finally, the paper concludes with a discussion of future work.

2. Background

In this section, we present a review of the existing works and discuss the relevant background information, including deep learning algorithms explored in this work.

2.1. BERT-Based Models

Bidirectional Encoder Representations from Transformers (BERT) is a neural network architecture for natural language processing (NLP) tasks, developed by Devlin et al. in 2018 [4]. BERT-based models are pre-trained on large amounts of text data and can be fine-tuned for a wide range of NLP tasks, such as sentiment analysis, named entity recognition and question answering. BERT has shown state-of-the-art results on various benchmark datasets, and its architecture can be adapted for various downstream tasks with relatively little additional training data.

BERT-based models have been effective in Vision-and-Language Navigation (VLN). VLN involves training an agent to navigate a virtual environment based on natural language instructions. To this end, several researchers have explored BERT-based models. Hong et al. [5] proposed a recurrent vision-and-language BERT model called VLN BERT. They address the difficulty of adapting the BERT architecture to the partially observable Markov decision process in VLN by equipping the model with a recurrent function to maintain cross-modal state information for the agent. Their experiments demonstrate that VLN BERT achieves state-of-the-art results and can be used for merging the learning of VLN with other vision-and-language tasks. They also show that VLN BERT with recurrence is capable of VLN and referring expression (REF) multitasking. For a comparative study on various BERT-based models for VLN, readers are encouraged to refer to the following works [6,7].

BERT-based models have been effectively used for video captioning. For example, Ye et al. [8] proposed a hierarchical modular network for video captioning that learns to bridge video representations and linguistic semantics from three levels. A transformer encoder–decoder architecture and pre-trained 2D and 3D CNNs extracted context and motion features from video footage. The authors also use SBERT, a variation of BERT designed for sentence embeddings, to compute entity embeddings from captions and supervise their entity module using a distance cost between the entity embeddings and the language head outputs. The applications of BERT-based models extend to text augmentation [9], text-to-image generation [10], and text-to-speech synthesis [11].

2.2. Previous Work

Deep learning has been effectively used in related Arabic arts applications, including calligraphy style recognition and melody classification [12]. A review of the literature in the context of Arabic text emotion and poetry classification is presented next. Abdullah et al. [13] proposed the Sentiment and Emotion Detection in Arabic Text (SEDAT) system, which categorized Arabic text into different emotion categories using emotion intensities. The intensities ranged from 0 to 1 (i.e., least to most intensity), with four stages (0 representing no emotion, 1 as low emotion, 2 for moderate emotion, and 3

representing high emotions). The experiments were performed on Arabic tweets and English-translated tweets. The first sub-model combined the Arabic and English tweets and extracted a 4908-dimensional vector containing Arabic lexicon, emoji, document, and sentiment features. The second experiment set used only the Arabic tweets and applied AraVec https://github.com/bakrianoo/aravec (accessed on 21 April 2023) to extract a 300-dimensional embedding vector. The first sub-model consisted of three dense layers with Stochastic Gradient Descent (SGD) optimizer and Mean Square Error (MSE) loss function. The second sub-model was a Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) network, where the CNN was used first for feature extraction and the LSTM for sequence prediction.

SGD optimizer was also applied for the second sub-model. Overall, the second sub-model performed better than the first for anger, joy, fear, and sadness. Furthermore, the model performed best for anger and fear emotions using Modern Standard Arabic (MSA), for joy using the Egyptian dialect, and for sadness using the Gulf dialect. A machine learning approach for subjectivity and sentiment analysis was proposed in [14]. Text from Penn Arabic Treebank was used to classify whether the text was subjective or objective.

Moreover, a second classifier was implemented to categorize whether the text was of positive or negative sentiment. Features including unique words, N-Grams, and adjectives were used to train a Support Vector Machine (SVM) model. The reported F1-score for subjectivity was 0.72, and 0.956 for sentiment. A classification system for emotion in tweets that can help different entities to analyze reactions regarding political, economic, or company decisions was proposed in [1]. The authors crawled and used 11,503 Arabic tweets using Tweepy API, and the considered emotions were anger, fear, disgust, joy, sadness, and surprise. Two annotators provided labels for emotion intensities for the tweets. The annotation outputs were discretized and turned into Boolean values, where 0 represented no emotion, and 1 represented the classified emotion. The Random Forest (RF) classifier had the highest accuracy, recall, and F1-score, followed by the decision tree and K-Nearest Neighbor (KNN). RF also had the lowest hamming loss of 0.357.

Alswaidan and Menai [2] introduced a hybrid model for emotion recognition in Arabic text to improve human–computer interaction and to help in other fields such as psychology and robotics. The model consisted of two elements. The human-engineered features (HEF) and deep features (DF) models. The first architecture manually extracted linguistic, lexical, syntactic, and semantic features. In contrast, the second architecture extracted and combined embeddings from emoji2vec, AraVec-CBOW, AraVec-SkipGram, and FastText pre-trained models to create a 1500-dimensional vector. Then, the features of both models were concatenated and passed to a three-layer deep neural network. Three datasets were used: Arabic tweets, Egyptian dialect tweets, and Iraqi Facebook posts. Three models were tested, namely HEF, DF, and hybrid. The training was done using 10-fold cross-validation. The results show that the hybrid model performed best when combining the hourglass of emotions (HGE) and lexical features from the HEF with the DF model.

Using Twitter data, Rabie and Sturm [15] and Al-Hagery et al. [3] tested different machine-learning methods for Arabic social media emotion classification. According to [3,16], analyzing text emotions helps in business development, content creation, and extracting opinions on current events and human interests. Rabie and Strum [15] tested the crawled tweets on sequential minimal optimization algorithm (SMO) and Naïve Bayes (NB) algorithms. The results show that SMO provided a 5.4% increase in the classification of the six emotions (anger, disgust, fear, happiness, sadness, and surprise) compared to NB. Al-Hagery et al. [3] experimented with three different algorithms: NB, Linear Regression (LR), and SVM, and two different feature combinations: Bag-of-Words (BoW) with N-grams, and TF-IDF with N-grams. The results indicate that the first feature combination provided the best performance, and SVM had the best results, followed by LR and then NB. Unlike the previous research that focused on detecting emotions, [16] focused on analyzing the sentiments of Arabic text (i.e., tweets). The authors used Word2vec continuous BOW for embedding the words. Six single classifiers were tested against six ensemble models.

Further, Synthetic Minority Over-sampling Technique (SMOTE) technique was used in training to solve the imbalanced dataset problem by creating synthetic samples of the minority classes. Overall, the outcome showed that SMOTE increased the performance, and that the stacking ensemble method provided the best results.

Some current work has been focused on the classification of Arabic poetry based on the era of the poems. For example, Gharbat et al. [17] explored four machine learning algorithms, including logistic regression, decision trees, RF, and SVM, to classify poems into Abbasid and Andalusian eras. A total of 10,895 hemistiches (a part of the poem line) were used from an Arab poetry website to implement binary classifiers. After the preprocessing steps, including the removal of prepositions and tokenization, the authors performed rooting and stemming by obtaining relevant features. Next, the dataset was split into training and test sets (without cross-validation) and then used to train the models. The highest accuracy of 70.55% was obtained using SVM.

Similarly, [18] explored various machine learning algorithms to classify Arabic poems from Pre-Islamic, Ummayad, Abbasid, and Andalusian eras. A total of 30,866 poems from a Kaggle dataset were utilized for model training. A word tokenizer was used to tokenize and text-mine all the words in the corpus. Moreover, the N-gram tokenizer was used to improve the results. Multinomial Naïve Bayes (NB) scored the best results, with 70.2% accuracy and a 0.68 F1-score. The authors in [19] utilized a deep learning approach to classify Arabic poems into five eras: Modern, Andalusian, Abbasid, Umayyad, and Pre-Islamic. The dataset was scraped using the Adab website, and contained over 60,000 poems. Unlike the previously mentioned approaches, they utilized word embeddings from FastText with a 200-dimension vector, and the minimum and maximum length of character N-grams were set to 3 and 6, respectively. A CNN classifier was utilized to predict two, three, and five different eras. For the five-era classification, the highest reported F1 score was 0.796.

The classification of classical Arabic poetry into various categories using NB was proposed by Mohammad [20]. The authors collected the dataset using well-known Arabic literature books containing poetry belonging to eight classes: Hekmah, Retha'a, Ghazal, Madeh, Heja'a, Wasef, Fakher, and Naseeb. Besides extracting the root words from the poem text, the author also considered the rank and order of the words. However, the dataset size in this work was very small, and the test set only contained 20 poems. The accuracy obtained was only 55%.

Moreover, no baseline model for comparison was presented apart from the proposed NB model. Similarly, Alsharif et al. [21] used machine learning to classify Arabic poems into four emotion categories: Retha, Ghazal, Fakhr, and Heja. The dataset contained 1231 poems and was used to train NB and SVM classifiers. The first nonlinear function returns the top unigrams with the most occurrences throughout the texts. The second function returns the top unigrams with the most mutually deducted occurrences in the texts. The NB model outperformed SVM with an F1-score of 0.66 using 1200 unigrams. Ahmed et al. [22] conducted poem classification in four different classes: love, Islamic, political, and social poems. The authors crawled poems from a website and applied preprocessing, which includes stemming tokenization and removing non-Arabic words. Three machine learning classifiers were used: SVM, NB, and Linear Support Vector Classification (SVC). The best results were provided using the linear SVC, with an average of 0.72, 0.47, and 0.51 for precision, recall, and F1-Score. The love poems had the best classification performance for the linear SVC and NB, while Islamic poems performed the highest using SVM. Table 1 summarizes the existing work in Arabic emotion and poem classifications.

Table 1. Existing Works in Arabic Text Emotion and Poetry Classification.

Source	Application	Dataset	Features	Algorithm	Results
[13]	Detect sentiment and emotions in Arabic tweets	SemEval Task-1: Affect in Tweets	1st submodel (Arabic tweets + English translated tweets): AffectiveTweets (142D), Doc2Vec (600D), Arabic Features (5D), DeepEmoji (64D), Unsupervised Learning sentiment features (4096D), Emoji Feature (1D) 2nd submodel (Arabic tweets only): 300D Aravec word embedding	CNN-LSTM	Emotion classification (Spearman correlation): 0.569
[14]	Subjectivity and sentiment analysis	2855 Arabic annotated sentences from Penn Arabic Treebank	Domain labels, unique words, N-Grams, and adjectives	SVM	0.72 F1 score for subjectivity and 0.956 F1 score for sentiment
[1]	Classify emotions in Arabic tweets	Crawled 11,503 Arabic tweets	Bag-of-Words (BOW) and Term frequency-inverse document frequency (TF-IFD)	DT, RF, KNN	Accuracy: 0.664 Precision: 0.674 Recall: 0.857 F1-score: 0.826 (Best scores using RF)
[2]	Emotion recognition in Arabic text	AETD, IAEDS, and SemEval	TF-IDF of character-grams (1–10 characters), TF-IDF of uni-grams Lexical sentiment features, Lexical emotion features Syntactic features: TF-IDF of POS Semantic features: TF-IDF of semantic meanings Deep features (embeddings): Emoji2vec (300D), GloVe (300D), AraVec-CBOW (300D), AraVec- SkipGram (300D), FastText (300D)	DNN (HEF) CuDNNLSTM + CuDNNGRU Hybrid HEF+DF	HEF+DF using IAEDS dataset: Accuracy: 87.2% Precision: 0.69 Recall: 0.60 F1-score: 0.64
[15]	Detection of emotions in Arabic social media content	Crawled 1605 Arabic tweets	Word–emotion lexicon built using Weka—BestFirst algorithm and manually selected emotion related words	SMO, NB, and simple search and frequency (SF) algorithm	Average precision: 0.74 Average recall: 0.64 Average F1-score: 0.65 (Best scores using SF)
[3]	Classification of emotions in Arabic tweets	Crawled 3171 Arabic tweets	BoW, TF-IDF, N-grams	SVM, NB, LR	Accuracy: 82.43% F1-score: 0.83 (SVM)
[16]	Sentiment analysis for imbalanced Arabic text dataset	Syria tweets dataset	Word2vec-CBOW 300D	SGD, SVC, LR, Gaussian NB, KNN, DT Ensemble including RF, voting, stacking	Accuracy: 85.28% F1-score: 63.95% (Stacking)
[17]	Arabic poetry era classification	10,895 hemistiches from *Adab* website belonging to Abbasid and Andalusian eras	Words in poetry text, rooting and stemming	SVM, logistic regression, RF, DT	Accuracy: 70.5% (SVM)

Table 1. *Cont.*

Source	Application	Dataset	Features	Algorithm	Results
[18]	Arabic poetry era classification	30,866 poems belonging to Pre-Islamic, Umayyad, Abbasid, and Andalusian eras from Kaggle	Word tokenizer and N-gram tokenizer	Multinomial NB	Accuracy: 70.21%, F1-Score: 0.68
[19]	Arabic poetry era classification	86,061 poems categorized into five eras collected from *Adab* website	FastText word embeddings Skipgram model (200D)	CNN	F1-Score: 0.796 (on five eras classification)
[20]	Classical Arabic poetry classification	Poems used from well-known books into eight categories: Hekmah, Retha'a, Ghazal, Madeh, Heja'a, Wasef, Fakher, Naseeb	Root words from poem text, order and rank of words	NB	Recall: 0.6000 Precision: 0.7500 Accuracy: 55%
[21]	Classify Arabic poetry emotion	1231 poems from online website in four categories: Retha, Ghazal, Fakhr, Heja	Top K Unigrams	NB, SVM	Precision: 0.73 Recall: 0.66 F1-score: 0.66 (NB)
[22]	Classification of modern Arabic poetry	Arabic poetry dataset collected from websites	Mutually deducted occurrence	SVM, NB, Linear SVC	Precision: 0.72 Recall: 0.47 F1-score: 0.51 (SVM)

2.3. Deep Learning Algorithms

Traditional machine learning (ML) techniques require human experts' manual extraction of features. Unlike ML, deep learning (DL) methods automatically learn and detect the required input features and optimize them. This work explores three types of DL models: convolutional neural networks, recurrent neural networks, and transformer models. Since the input of our model was a sequence of a vector of tokens representing text, 1D-CNN was one reasonable model. The bidirectional gated recurrent units (GRU) and long-short-term memory (LSTM) cells were used for the recurrent networks. Finally, the AraBERT transformer model [23] was also considered. Four popular metrics, namely, accuracy, precision, recall, and F1 score, were employed for evaluation.

A more complex learner, such as a deep learning model, often requires a larger number of parameters to be trained, which can result in a significant increase in computational resources for model training. As the number of parameters in a model increases, so does the computational cost of training, which can result in longer training times and higher memory usage. Additionally, more complex models often require larger datasets to avoid overfitting, which can also increase the processing time for training. Therefore, it is important to consider the trade-off between model complexity and practical considerations, including computational efficiency. In some cases, a simpler model, such as a machine learning model, may be more appropriate, due to its lower computational requirements and faster training times. However, in other cases, the higher accuracy achieved by a more complex model may outweigh the additional computational cost. In this context, we present a comparison of several machine learning and deep learning models.

3. Methodology

Because the poems are written in standard language and not in various dialects, existing libraries for pre-processing the Arabic text were used. Tnkeeh and tkseem Arabic NLP libraries [24] and AraBERT [23] were used to apply Arabic-related functions, such as detecting the different diacritics and tatweel (elongation), and other general methods that involve handling special characters and whitespaces, limiting text length, and tokenization.

One difficulty in solving this problem stems from the fact that there were no available datasets for Arabic poems with emotional labels. Therefore, a new Arabic poem-emotion dataset was created with three emotions: sadness, joy, and love. Furthermore, the dataset was implemented on four different models, which are 1D-CNN, bidirectional RNN, CNN-LSTM, and the AraBERT transformer model, which were compared with five traditional Machine Learning (ML) classifiers and three ensemble methods. The convolutional networks are used in different classification problems by applying a grid on top of the input matrix or vector to extract various features. Meanwhile, recurrent networks are used for predicting sequential data. AraBERT is an Arabic pre-trained model based on Google's BERT architecture, consisting of bidirectional encoders, attention heads, and around 110 million parameters [23]. In addition, the model is trained on 70M sentences comprising around three billion words [23]. To the best of our knowledge, no Arabic poetry emotion classification using deep learning has been previously explored.

3.1. Dataset Collection

The dataset used by [21] to classify emotions in Arabic poems contained only 1231 poems, and traditional machine learning was used for classification. However, deep learning generally requires larger datasets. Therefore, a poem emotion dataset for classification was constructed. A web scraper to collect poems with their corresponding emotions from Al Diwan [25] https://www.aldiwan.net (accessed on 21 April 2023) website was developed to construct the dataset. This website was selected because it was the only source where poems were categorized under several emotional categories. Another source, like Kaggle, contained Arabic poetry datasets classified by the era of the poems only. For the sadness emotion, sad and lament poems were combined, and for the love emotion, romantic poems were used.

Consequently, 9452 poems belonging to the three emotion classes were collected, as summarized in Table 2. Sample Arabic text and corresponding translation are also presented in Table 2 for the three categories. The dataset can be accessed from [26].

Table 2. Poem Dataset Example and Summary.

Arabic Poem Example	Emotion	Total Poems
If they asked about my wellbeing, I tried to make excuses, but I can't I say I'm fine but it's all words on the tongue And God knows what's deep inside and what the eyes hide	Sad	4064
The heart gets what it wants from love, and the body gets its share of disease And if you love you find the ordeal of who loves, death I am amazed from my heart that never tires from longing and blaming hurts it	Love	3443
A face as if the full moon borrowed its light and shine from it And I saw on him a garden that made me and everyone else look at him	Joy	1945
		9452

3.2. Deep Learning Approach

The input dataset was pre-processed similarly for all deep learning methods. The emotion labels were converted into numerical representations (i.e., sad—0, love—1, joy—2), the dataset was shuffled, and Tnkeeh and tkseem Arabic NLP libraries [24] were used for text cleaning and tokenization. After cleaning the text, a vocabulary index was created for each unique word. The indexes, or tokens, were then applied to the training and testing sets to replace each word with its corresponding token. The maximum length for each poem (i.e., the maximum number of tokenized words in the sample) was set to 75, and max padding was used. Finally, the integer labels were converted into a binary 3D matrix.

For the 1D-CNN model, the input vectors were embedded into a 32-dimensional latent space. One convolutional layer was used with 100 filters of size 5, padding, and ReLU activation function, followed by a dropout layer [27] with a 0.5 rate and a max-pooling layer with a pool size of 2. The output was flattened and fed to two dense layers. The first layer had 256 units with ReLU [28] activation, 0.5 dropout rate, and batch normalization. The second layer was a classification layer with three neurons corresponding to the emotions and SoftMax function. Adam [29] optimizer was applied, and the model used categorical cross-entropy loss.

The bidirectional RNN consists of an embedding layer with 16-dimensions, two 32-GRU bidirectional layers, and a dropout layer with a 0.2 rate. GRU consisted of two gates, reset (r) and update (z), that provide a long memory for the network. The reset gate selects the information from the previous hidden state (h_{t-1}) to forget, and the update gate decides the relevant information to keep from the past state. The gates and the weight matrix (W) are used to calculate the current hidden vector. Equation (1) formulates the entire process:

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot tanh[W \cdot x_t + W \cdot r_t \cdot h_{t-1}] \quad (1)$$

A dense layer was added after the GRU layers with 64 neurons, ReLU activation, 0.5 dropout rate, and batch normalization. The classification layer was similar to the previous model.

The last method was a 1DCNN + LSTM model. The input was embedded into a 32-dimensional space and then passed to a convolutional layer with 32 filters of size 5, padding, and ReLU activation. Dropout of 0.1 and 1D max pooling with grid size two were used, followed by an LSTM layer with 32 cells and a dropout rate of 0.4. Similar to the GRU, the LSTM cell was created to solve the short-term memory problem. The LSTM cell has three gates, forget (f), input (i), and output (o), with a cell state (c_t). The cell state passes relevant information throughout the network. The gates decide what information is

removed or added to the cell state and leaked to the hidden vector. Equations (2) and (3) show the cell state and hidden vector for the timestamp t:

$$c_t = f.c_{t-1} + i.c \tag{2}$$

$$h_t = o.\tanh(c_t) \tag{3}$$

The output of the LSTM was passed to the dense network. The fully connected network was similar to the previous approach, but instead of a 0.5 dropout rate, it was set to 0.4.

3.3. AraBERT Model

As previously mentioned, AraBERT is an extension of Bidirectional Encoder Representations from Transformers (BERT) [4]. Instead of considering a sequence of text from left to right side or vice versa (single direction) in traditional approaches, BERT takes advantage of bidirectional flow, which provides a more sophisticated context representation. The configuration used in AraBERT consisted of 12 encoder blocks, 768 hidden dimensions, 12 attention heads, 512 maximum sequence lengths, and a total of ∼110M parameters [23]. This work utilized the second version of the AraBERT model, AraBERTv0.2-base. The data were pre-processed using the model functions for cleaning and tokenizing the text. The functions include removing tashkeel (i.e., diacritics) and tatweel, replacing elongation (i.e., repeated characters with a frequency of more than two) with two of the same character, and inserting whitespaces between non-Arabic/English digits or alphabets, brackets, and numbers or words [23]. The text was padded to a length of 256 characters. We summarize our methodology in Figure 1.

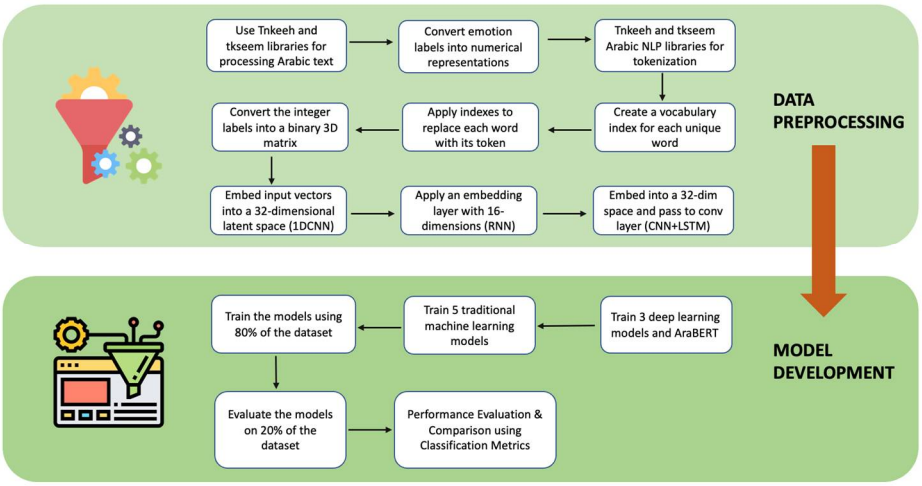

Figure 1. Methodology Summary.

4. Results

We first trained the models using 80% of the dataset in both approaches. Next, the models were evaluated on the test dataset, which constituted 20% of the dataset. The 80:20 split ratio provides an adequate number of samples for the training and testing phases, with 7562 and 1890 samples, respectively. Given that deep learning models require more training samples, we determined that a training size of 80% sample provided a reasonable tradeoff. This sample size helps reduce model overfitting and improves generalization on unseen data. The results from the various models are presented next.

4.1. Deep Learning Results

As discussed in the previous section, three deep learning architectures were experimented with and compared with ML and ensemble classifiers. The ML classifiers implemented are Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Multi-Layer Perceptron (MLP), Decision Tree (DT), and Gaussian Naïve Bayes (NB). The ensemble methods used are gradient boosting, adaptive boosting, and random forest. For all architectures, the split validation parameter was set to 0.2, which means that 20% of the training data would be used for validation. Figure 2 presents the confusion matrix on the test set using the 1D-CNN method. The maximum accuracy on the training set after 15 epochs was 97%. However, the model was overfitted because the accuracy on the test set was 62%. In fact, all three deep learning models showed similar overfitting behavior when testing, as summarized in Table 3. Attempting to resolve the overfitting issue by adding dropout layers and regularization parameters helped to a certain extent. The F1-score obtained on the test set was 0.62. The 1D-CNN could predict the sad poems most successfully (70% accuracy), whereas it was least accurate in joy poems (56%).

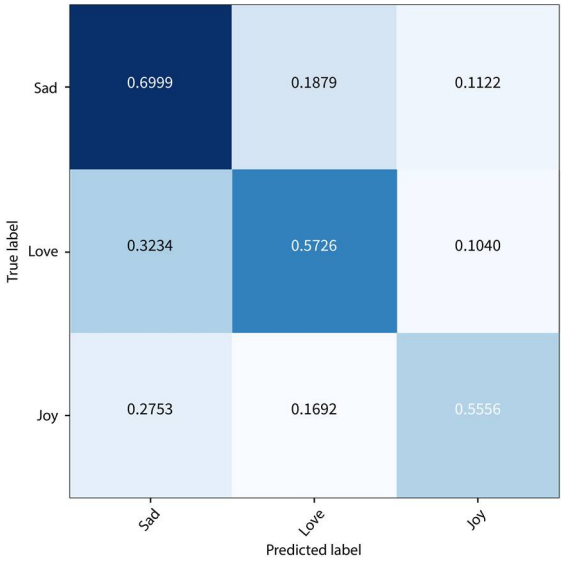

Figure 2. Confusion Matrix on the Test Set using 1D-CNN.

Table 3. Performance Comparison of the DL and ML Architectures.

Architecture	Train Accuracy (%)	Test Accuracy (%)	Testing F1-Score
1D-CNN	97.0	62.0	0.62
Bi-RNN (GRU)	98.0	63.0	0.62
1DCNN + LSTM	97.0	54.0	0.53
SVM	68.0	44.0	0.288
KNN (K = 5)	58.0	41.0	0.312
MLP	55.0	32.0	0.314
DT	100.0	37.0	0.343
Gaussian NB	40.0	37.0	0.349
Gradient Boosting	60.0	44.0	0.318
AdaBoost	45.0	42.0	0.322
Random Forest	100.0	45.0	0.309

The confusion matrix on the test set for the GRU-based bidirectional RNN is presented next in Figure 3. Using this approach, the accuracy on the test set was 63%, and the F1-score was 0.62. Interestingly, this approach predicted sad poems with 78% accuracy, although joy poems were inaccurate with 44% accuracy. Therefore, no significant improvement was obtained using this approach compared to the first method.

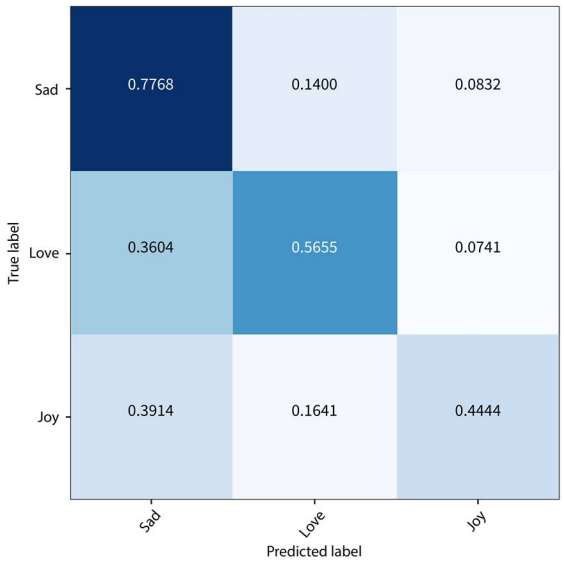

Figure 3. Confusion Matrix on the Test Set using Bidirectional RNN.

Next, a combination of 1D-CNN as well as LSTM layers was explored. The confusion matrix on the test set using this approach is presented in Figure 4. Overall, this approach resulted in the worst performance among the deep learning models, with an overall accuracy and F1-score of 54% and 0.53, respectively, on the test set. The model could not discriminate the joy poems from the sad and love ones, resulting in only 28% accuracy. Comparing the previous methods with the traditional classifiers proves the superiority of deep learning models, as shown in Table 4. The DT and random forest methods are clearly overfitting. Despite most traditional methods not overfitting, the results were inferior. The highest F1-score of 0.349 was achieved using the Gaussian NB. Further, the first two deep learning models increased the F1-score by 0.271.

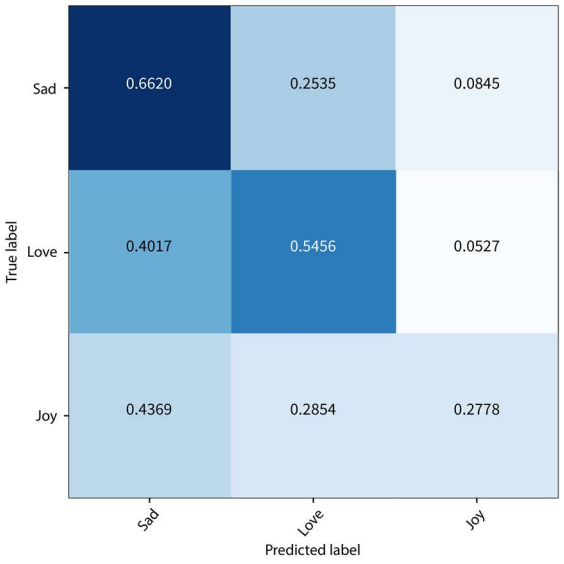

Figure 4. Confusion Matrix on the Test Set using 1DCNN + LSTM.

Table 4. Performance Comparison with Existing Arabic Poetry Classification.

Source	Classification	Accuracy (%)	F1-Score
[17]	Poem era	70.5	0.62
[18]	Poem era	70.2	0.68
[19]	Poem era	80.1	0.80
[20]	Classical poetry	55.0	0.67
[21]	Emotion	-	0.66
[22]	Modern poetry	-	0.51
Ours	**Emotion**	**76.5**	**0.77**

4.2. AraBERT Results

This section shows the results for the transformer-based AraBERT model. For consistency, 80% of the data for training and 20% for evaluation were used. The model training was stopped after three epochs, as training for longer caused overfitting. Overall, the AraBERT model could learn the inputs' general features without capturing the details, which prevents overfitting. The training and testing accuracies were 76.5%. The performance of this model was better than the previous ones. The outstanding performance of the pre-trained model is probably due to the large pre-trained corpus. Figure 5 presents the confusion matrix of the testing sample. As with the previous approaches, sad poems were classified most accurately (80%), with joy poems being the least accurate (67%). Subsequently, this is most likely due to the dataset having twice as many sad poms as joy poems. We obtained an overall precision, recall, and F1-score of 0.76, 0.77, and 0.77 on the test set, respectively.

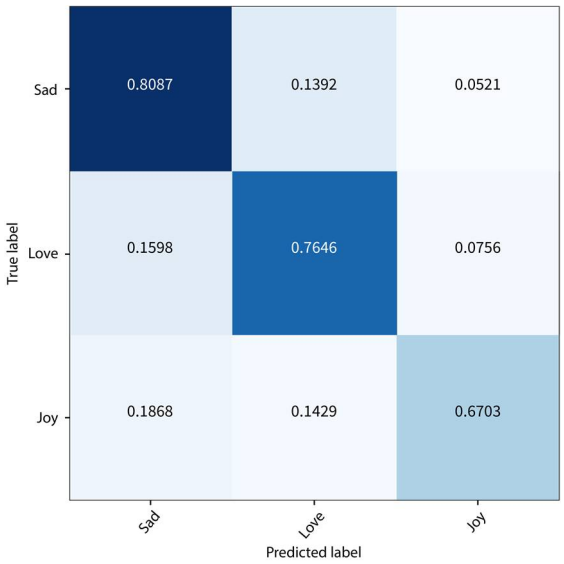

Figure 5. Confusion Matrix on the Test Set using AraBERT.

4.3. Comparison and Discussion

Looking at the results using the deep learning approach and the AraBERT model, a significant improvement using AraBERT can be observed. The improvement in F1-score was 24%, whereas the improvement in overall accuracy was 21%. Across all models, the probability of predicting the sad poems is highest while joy poems is the lowest, due to the dataset imbalance. Table 4 presents a performance comparison between the existing works in poem classification and the proposed work. The proposed work significantly outperforms all existing works utilizing machine learning approaches. The reported performance in [19], which utilized deep learning, was better than the work presented here. However, it must be noted that [19] utilized a different dataset than this work because the

problem they addressed was poem era classification. Across the same emotion classification problem, the proposed approach significantly improves upon the previous work [21].

5. Conclusions and Future Work

This work presents three deep learning models and the AraBERT transformer model for classifying Arabic poems based on emotion. A novel Arabic poetry dataset was introduced for the classification task. The AraBERT transformer model was superior when compared to the deep learning models. Compared to previous works, the proposed work outperformed the machine learning-based approaches for Arabic poem classification. As future work, the dataset should be expanded to contain other poem categories such as spiritual and nature. Moreover, the dataset should be balanced across all the categories for further improvements in prediction performance. Finally, the dataset could also be used for Arabic poetry generation using generative adversarial networks (GANs).

Author Contributions: Conceptualization, S.S. and N.A.R.; methodology, I.Z.; software, N.A.R.; validation, S.S. and N.A.R.; formal analysis, N.A.R. and S.S.; investigation, S.S. and N.A.R.; resources, I.Z.; data curation, S.S. and N.A.R.; writing—original draft preparation, S.S. and N.A.R.; writing—review and editing, I.Z.; visualization, S.S. and N.A.R.; supervision, I.Z.; project administration, I.Z.; funding acquisition, I.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Open Access Program at the American University of Sharjah, UAE.

Data Availability Statement: The labelled data is publicly available at https://github.com/SakibShahriar95/Arabic-Poem-Emotion (accessed on 9 March 2023).

Conflicts of Interest: The authors have no conflict of interest to declare. All co-authors have seen and agree with the contents of the manuscript, and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

References

1. Alzu'bi, S.; Badarneh, O.; Hawashin, B.; Al-Ayyoub, M.; Alhindawi, N.; Jararweh, Y. Multi-Label Emotion Classification for Arabic Tweets. In Proceedings of the 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 22–25 October 2019; pp. 499–504. [CrossRef]
2. Alswaidan, N.; Menai, M.E.B. Hybrid Feature Model for Emotion Recognition in Arabic Text. *IEEE Access* **2020**, *8*, 37843–37854. [CrossRef]
3. Al-Hagery, M.; Alassaf, M.; Al-kharboush, F. Exploration of the best performance method of emotions classification for arabic tweets. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *19*, 1010. [CrossRef]
4. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:181004805.
5. Hong, Y.; Wu, Q.; Qi, Y.; Rodriguez-Opazo, C.; Gould, S. Vln bert: A recurrent vision-and-language bert for navigation. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
6. Qiao, Y.; Qi, Y.; Hong, Y.; Yu, Z.; Wang, P.; Wu, Q. HOP+: History-enhanced and Order-aware Pre-training for Vision-and-Language Navigation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2023; *ahead of print*. [CrossRef]
7. An, D.; Qi, Y.; Huang, Y.; Wu, Q.; Wang, L.; Tan, T. Neighbor-view enhanced model for vision and language navigation. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021.
8. Ye, H.; Li, G.; Qi, Y.; Wang, S.; Huang, Q.; Yang, M.H. Hierarchical modular network for video captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
9. Atliha, V.; Šešok, D. Text augmentation using BERT for image captioning. *Appl. Sci.* **2020**, *10*, 5978. [CrossRef]
10. Zhou, Y.; Shimada, N. Generative adversarial network for text-to-face synthesis and manipulation with pretrained BERT model. In Proceeding of the 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Jodhpur, India, 15–18 December 2021.
11. Hayashi, T.; Watanabe, S.; Toda, T.; Takeda, K.; Toshniwal, S.; Livescu, K. Pre-Trained Text Embeddings for Enhanced Text-to-Speech Synthesis. In Proceeding of the 20th Annual Conference of the International Speech Communication Association INTERSPEECH 2019, Graz, Austria, 15–19 September 2019.
12. Shahriar, S.; Tariq, U. Classifying Maqams of Qur'anic Recitations using Deep Learning. *IEEE Access* **2021**, *9*, 117271–117281. [CrossRef]
13. Abdullah, M.; Hadzikadicy, M.; Shaikhz, S. SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 835–840. [CrossRef]

14. Abdul-Mageed, M.; Diab, M.; Korayem, M. Subjectivity and sentiment analysis of modern standard Arabic. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland Oregon, 19–24 June 2011; pp. 587–591.
15. Rabie, O.; Sturm, C. Feel the heat: Emotion detection in Arabic social media content. In Proceedings of the International Conference on Data Mining, Internet Computing, and Big Data (BigData2014), Washington, DC, USA, 17–19 November 2014; pp. 37–49.
16. Al-Azani, S.; El-Alfy, E.-S.M. Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text. *Procedia Comput. Sci.* **2017**, *109*, 359–366. [CrossRef]
17. Gharbat, M.; Saadeh, H.; Al Fayez, R.Q. Discovering The Applicability of Classification Algorithms With Arabic Poetry. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 453–458. [CrossRef]
18. Abbas, M.; Lichouri, M.; Zeggada, A. Classification of Arabic Poems: From the 5th to the 15th Century. In *New Trends in Image Analysis and Processing—ICIAP 2019*; Cristani, M., Prati, A., Lanz, O., Messelodi, S., Sebe, N., Eds.; Springer International Publishin: Cham, Switzerland, 2019; pp. 179–186.
19. Orabi, M.; Rifai, H.E.; Elnagar, A. Classical Arabic Poetry: Classification based on Era. In Proceedings of the 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), Antalya, Turkey, 2–5 November 2020; pp. 1–6. [CrossRef]
20. Mohammad, I. Naive bayes for classical arabic poetry classification. *Al-Nahrain J. Sci.* **2009**, *12*, 217–225.
21. Alsharif, O.; Alshamaa, D.; Ghneim, N. Emotion classification in Arabic poetry using machine learning. *Int. J. Comput. Appl.* **2013**, *65*, 16.
22. Ahmed, M.A.; Hasan, R.A.; Ali, A.H.; Mohammed, M.A. Using machine learning for the classification of the modern Arabic poetry. *TELKOMNIKA Telecommun. Comput. Electron. Control* **2019**, *17*, 2667. [CrossRef]
23. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-Based Model for Arabic Language Understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 12 May 2020*; European Language Resource Association: Luxembourg, 2020; pp. 9–15. Available online: https://www.aclweb.org/anthology/2020.osact-1.2 (accessed on 22 May 2021).
24. Alyafeai, Z.; Al-Shaibani, M. ARBML: Democritizing Arabic Natural Language Processing Tools. In Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), Online, 9 November 2020; pp. 8–13.
25. "Al Diwan: Encyclopedia of Arab Poetry". Available online: https://www.aldiwan.net (accessed on 20 May 2021).
26. Shahriar, S. Arabic-Poem-Emotion. GitHub. Available online: https://github.com/SakibShahriar95/Arabic-Poem-Emotion (accessed on 6 September 2021).
27. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
28. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 807–814.
29. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:14126980.