01010
01010
01010
information

MDPI

Article

The Evolution of Language Models Applied to Emotion Analysis of Arabic Tweets

Nora Al-Twairesh

Department of Information Technology, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; twairesh@ksu.edu.sa

Abstract: The field of natural language processing (NLP) has witnessed a boom in language representation models with the introduction of pretrained language models that are trained on massive textual data then used to fine-tune downstream NLP tasks. In this paper, we aim to study the evolution of language representation models by analyzing their effect on an under-researched NLP task: emotion analysis; for a low-resource language: Arabic. Most of the studies in the field of affect analysis focused on sentiment analysis, i.e., classifying text into valence (positive, negative, neutral) while few studies go further to analyze the finer grained emotional states (happiness, sadness, anger, etc.). Emotion analysis is a text classification problem that is tackled using machine learning techniques. Different language representation models have been used as features for these machine learning models to learn from. In this paper, we perform an empirical study on the evolution of language models, from the traditional term frequency–inverse document frequency (TF–IDF) to the more sophisticated word embedding word2vec, and finally the recent state-of-the-art pretrained language model, bidirectional encoder representations from transformers (BERT). We observe and analyze how the performance increases as we change the language model. We also investigate different BERT models for Arabic. We find that the best performance is achieved with the ArabicBERT large model, which is a BERT model trained on a large dataset of Arabic text. The increase in F1-score was significant +7–21%.

Keywords: pretrained language models; BERT; emotion analysis; Arabic

check for updates

Citation: Al-Twairesh, N. The Evolution of Language Models Applied to Emotion Analysis of Arabic Tweets. *Information* **2021**, *12*, 84. https://doi.org/10.3390/info12020084

Academic Editor: Yannis Korkontzelos

Received: 4 January 2021
Accepted: 15 February 2021
Published: 17 February 2021

1. Introduction

Language is complex and processing it computationally is not straight forward. The basic building block of language is words, in natural language processing (NLP) we need to convert words into numerical format to compose a suitable representation that can help machines to understand language. Thus, to represent language, the vector space model is used, hence words are represented as vectors of numbers. The different approaches for constructing these vectors are called language representation models or language models (LM). Language models in NLP have evolved from simple frequency counts such as bag of words, n-grams, and term frequency–inverse document frequency (TF–IDF), to more sophisticated representations that utilize neural networks to learn features automatically in an unsupervised way from large datasets, such as word2vec [1], GloVe [2], and fastText [3]. However, while major advances were achieved with these early models, they still lacked contextualized information.

Recently, the NLP field has witnessed a major breakthrough with the development of the transformer architecture [4] that led to the innovation of pretrained language models (PLM) such as bidirectional encoder representations from transformers (BERT) [5], OpenAP GPT [6], RoBERTa [7], XLNet [8], ALBERT [9], and ELECTRA [10] to name a few. These new LMs are trained on massive unlabeled datasets and learn word representations that are contextual, i.e., each word has different representations depending on the context it appears in, hence, capturing uses of words across varied contexts. These LMs are called contextual embeddings to distinguish them from the previous non-contextual embeddings (word2vec, GloVe, fastText) which only have one representation for words regardless of context.

These PLMs, coupled with the notion of transfer learning—a technique where instead of training a model from scratch, models pre-trained on a large dataset are used and then fine-tuned for specific natural language tasks—have led to significant performance gains in almost every NLP task [5,11].

Emotions are a vital component of language and are known to be complex and nuanced [12]. Emotion analysis refers to the task of NLP that detects the emotion of the writer of the text being analyzed. Several categorization models of emotions exist such as [13,14]. Ekman [13] introduced a six-category model while Plutchik [14] proposed an eight-category model (joy, sadness, anger, fear, trust, disgust, surprise, and anticipation) that are shown in Figure 1. Emotion detection has several applications in different fields such as in public health [15], e-commerce [16], and politics [17,18].

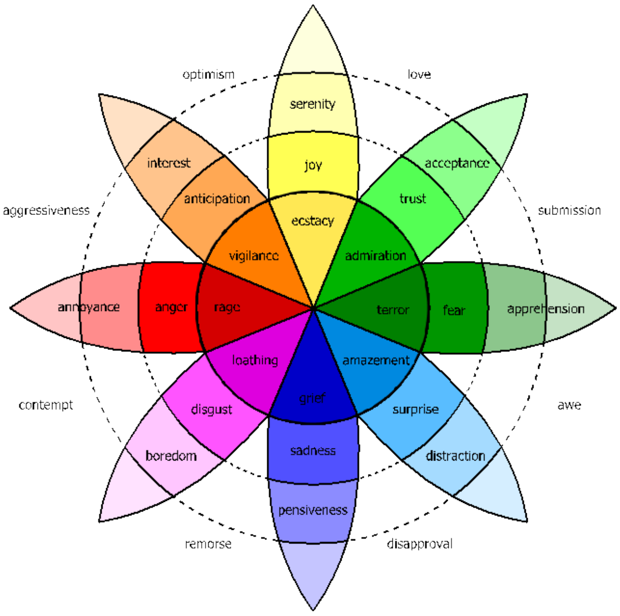


Figure 1. Plutchik wheel of emotion.

Compared to sentiment analysis, emotion analysis has not received the same attention from the research community. This is mainly due to the lack of labeled datasets [12,19]. This also applies to the Arabic language which is a low resourced language. However, recent attempts have emerged that provide labeled data for emotions in the Arabic language such as [19–22]. Previous work on emotion analysis for Arabic text ranged from traditional machine learning approaches to recent deep learning approaches. However, to the best of our knowledge, no work has been presented that investigates transfer learning and pretrained language models on emotion analysis of Arabic text. Therefore, in the paper we propose to study the evolution of language models by applying it to the emotion classification task. First we use the traditional TF–IDF for emotion analysis then we turn to more sophisticated word embedding word2vec, and finally we use the recent state-of-the-art pretrained language model BERT in the different versions available for Arabic: AraBERT [23], ArabicBERT [24], MultiDialect ArabicBERT [25].

The contributions of this paper are as follows:

- Studying the evolution of language models for Arabic: a low-resource language.
- Developing classification models for an under resourced NLP task: emotion analysis.
- Analysis of different BERT models for Arabic.

This paper is organized as follows: in Section 2 background and related work are overviewed. In Section 3 the methodology, dataset, and experiments are presented. The results of the experiments and discussion are given in Section 4. The conclusion and future directions are presented in Section 5.

2. Background and Related Work

Language representation models: Machine learning (ML) algorithms rely on features extracted from the training data to work efficiently and produce accurate models. Data and features fed into ML models have to be represented numerically. In the case of images, videos, or sounds, converting them into numerical form is straight forward. Images are converted to matrices of pixels, where at each cell the intensity of the corresponding pixel in the image is stored. Videos are similar, where a video is a collection of frames and each frame is an image. As for sound or speech, they are constituted of waves, hence the amplitude of a sound wave at fixed time intervals is used to represent speech mathematically. However, for textual data, converting it to a numerical form is not straight forward due to the complexity of language. Thus, it is an active research area that has received much attention from the NLP research community. This area is known as language or text representation. Language is complex, as it consists of different knowledge blocks: phonemes (speech and sound), morphology (words: morphemes and lexemes), syntax (phrases and sentences), and semantic (meaning and context). The basic building block of language is words. Words are composed of different morphemes and lexemes, are used to compose phrases and sentences, and have different meanings according to the context they appear in. All these different knowledge blocks have to be considered when we want to convert words into numerical format to compose a suitable representation that can help ML models to understand language and perform better on the various NLP tasks and applications. Thus, to represent language, the vector space model is used, hence words are represented as vectors of numbers. The different approaches for constructing these vectors are called language representation models or language models (LM).

Early language representations depended merely on frequency counts of words such as bag of words, n-grams, and TF–IDF. Although they are simple to calculate and have made progress in text classification tasks, these representations suffer from sparsity and generalization issues, especially with words that have different senses. Moreover, these traditional language representation models, are not sufficient to be used alone in text classification and are always augmented with hand crafted features of the task at hand. Manual feature extraction is a tedious task and consumes time and effort.

This paved the way for introducing dense vectors such as word2vec [1], GloVe [2], and fastText [3]. These neural language models have the advantage of being easier to include as features in machine learning systems, and they generalize better and help avoid overfitting because they contain fewer parameters than sparse vectors of explicit counts. They do not require any manual feature engineering (hand-derived features) and are learned automatically from text using a shallow neural network. Hence, in order to avoid the extensive human effort of feature design, recent research in NLP has focused on representation learning: ways to learn features automatically in an unsupervised way from the input.

Although these representations have improved performance on text classification tasks, they still suffer from several limitations. First, they are static, shallow, and do not depend on context, i.e., each word has only one vector representation no matter what the meaning. The second issue is the out-of-vocabulary problem: words that did not appear in the dataset do not have any representation.

Nonetheless, the year 2017 has witnessed a boom in the field of NLP with the advent of transformers [4]. Transformers consist of several layers of encoders and decoders which encapsulate multi-head attention components among other components. The previous recurrent neural networks (RNN) had the limitation of looking at the past words of a sequence, however, transformers overcome this limitation by looking at all the words surrounding the target word, and giving more weight to important words—this is known as *self-attention*. Therefore, each word is represented with respect to its context, thus words have more than one vector representation according to the respective meaning. This is in contrast to previous word embeddings (word2vec, GloVe, fastText) where each word had only one vector representation no matter what the context or meaning was. Additionally, transformers have the ability to process text in parallel rather than sequentially, thus improving execution speed.

Transformers coupled with the notion of transfer learning have led to the recent rise of pretrained language models (PLM). Transfer learning refers to the concept where the knowledge learned while solving one problem is applied to a different but related problem. Using a large-scale unannotated dataset and the transformer architecture, general purpose language models have evolved. These models learn universal language representations which lead to better performance and speed up convergence on the target task. The fine-tuning procedure follows the pre-training step, and in particular refers to the fine-tuning of downstream NLP tasks. In this procedure, a classification layer is added that computes the label probabilities using the standard softmax. These massive pre-trained language models, with billions of parameters learned from essentially all the text published on the internet, have improved the state-of-the-art on nearly every downstream natural language processing task, including question answering, conversational agents, and sentiment analysis, among others [11].

The transformer's basic architecture consists of blocks of an encoder and a decoder; the different PLMs use either the encoder alone or decoder alone or both by stacking several encoders/decoders which determines the size of the model. For example, the bidirectional encoder representations from transformers (BERT) model uses the encoder alone and has two different sizes: base which stacks 12 encoders and large which stacks 24 encoders. The BERT model has two objectives: masked language model (MLM) where 15% of the words in the dataset are masked and the model has to predict them given the previous and next word, and next sentence prediction (NSP) where the model given a sentence predicts the next sentence.

Ever since BERT was introduced, more language understanding models have been developed such as RoBERTa [7], XLNet [8], ALBERT [9], and ELECTRA [10], which improved performance by exploring different pretraining methods, modified model architectures, and larger training corpora.

Arabic language models: Arabic word2vec was first introduced by [26], where AraVec was released. It provides six different word embedding models learned from three different Arabic genres, namely, World Wide Web pages, Arabic Wikipedia articles, and tweets, using the word2vec model. For each genre, a continuous bag of words (CBOW) model and a skip-gram model were provided. The size of the dataset of tweets used to train the word2vec model was 77M tweets.

As for contextual language models, for non-English languages, a multilingual BERT was released which was trained on Wikipedia dumps of 100+ languages, Arabic being one of them. However, pre-training monolingual BERT for non-English languages has given better performance than the multilingual BERT. Therefore, Antoun et.al [23] proposed AraBert which is a monolingual version of BERT for the Arabic language. There are two versions of the model, AraBERTv0.1 and AraBERTv1, with the difference being that AraBERTv1 uses pre-segmented text where prefixes and suffixes were split using the Farasa segmenter [27] which is an Arabic specific text-segmenter. The model was trained on ~70 M sentences or ~23 GB of Arabic text with ~3B words. The training corpora are a collection of publicly available large scale raw Arabic text including Arabic Wikidumps, the 1.5 B words Arabic Corpus, the OSIAN Corpus, Assafir news articles, and four other manually crawled news websites. AraBERT was evaluated on several NLP tasks such as sentiment analysis, question answering, and named entity recognition and reported better performance on these downstream tasks.

Similarly, Safaya et al. [24] proposed ArabicBERT, which expands the previous AraBERT in the size of the corpora it was pretrained on. The models were pretrained on ~8.2 billion words from the Arabic version of Open Super-large Crawled ALMAnaCH19 coRpus

(OSCAR), a recent dump of Arabic Wikipedia, and other Arabic resources which sum up to ~95 GB of text. There are four versions of ArabicBERT according to the size of the architecture: mini, medium, base, and large. The details of the architectures of these four versions of ArabicBERT and the previous AraBERT are shown in Table 1. ArabicBERT was used for the downstream task of hate speech detection, where the authors participated in Subtask-A of the shared task of multilingual offensive language identification [28].

Table 1. Architecture details of AraBERT and ArabicBERT.

	AraBert	**ArabicBERT Mini**	**ArabicBERT Medium**	**ArabicBERT Base**	**ArabicBERT Large**
Hidden Layers	12	4	8	12	24
Attention heads	12	4	8	12	16
Hidden size	768	256	512	768	1024
Parameters	110 M	11 M	42 M	110 M	340 M

Moreover, the authors in [25] further pre-trained ArabicBERT Base on 10M tweets written in different Arabic dialects, for three epochs. They called this model the multi-dialect-Arabic-BERT. This new model was used for dialect identification and the authors won the first place in the nuanced Arabic dialect identification (NADI) Shared Task 1 [29].

To the best of our knowledge, these are the different monolingual BERT models available for the Arabic language. Therefore, we use these three models and their versions in our experiments.

After presenting this brief review of how language representation models have evolved, in this paper we aim to show empirically how the performance of these models has improved the performance of text classification tasks by experimenting on an under-resourced NLP task: emotion analysis. In the following section we review the related work on emotion analysis.

Emotion analysis: The task of emotion analysis has not received as much attention as sentiment analysis—this is true for all languages and is more evident in low-resourced languages such as the Arabic language. In this section, we review the related work on both English and Arabic emotion analysis.

In SemEval 2018 [19], a new task on affect in tweets was organized. The task covered both emotion and sentiment analysis with more emphasis on emotions. A new dataset of tweets was curated by collecting tweets that convey emotion in three languages: English, Arabic, and Spanish. There were five subtasks which are: (1) Emotion intensity regression (EI-reg), (2) emotion intensity ordinal classification (EI-oc), (3) valence (sentiment) regression (V-reg), (4) valence ordinal classification (V-oc), and (5) emotion classification (E-c). The first and second subtasks were on the intensities and classifications of four basic emotions: anger, fear, joy, and sadness. The third and fourth subtasks were on sentiment intensity and classification. The last subtask was on emotion classification over 11 emotions commonly expressed in tweets.

The Arabic query terms used in collecting the tweets were translated from the English query terms used for collecting the English dataset, also word embeddings trained on an Arabic tweet corpus were used to find synonyms of the translated query terms. The same emojis used for collecting the English tweets were also used to collect Arabic tweets. A total of 550 Arabic query terms and emojis were used to poll the Twitter API. Then for each of the four emotions (anger, fear, joy, sadness), 1400 tweets were randomly selected to form the EI-reg datasets. The same tweets were used for building the EI-oc datasets. For each of the four emotions E, the 0 to 1 range is partitioned into the classes: no emotion, low emotion, moderate emotion, and high emotion.

For the Arabic dataset, 13 teams competed, the teams that achieved the highest results mostly used deep learning approaches such as convolutional neural networks (CNN), long short-term memory (LSTM), and bidirectional LSTM (Bi-LSTM), and some used traditional machine learning approaches such as support vector machine (SVM) with features such as sentiment and emotion lexicons or word embeddings. Out of the 13 participants, only Badaro et al. [30], Mulki et al. [31], and Abdullah and Shaikh [32] submitted a paper describing their systems. Badaro et al. [30] used traditional machine learning (SVM, ridge classification (RC), random forests (RF), and an ensemble of the three) with features such as n-grams, affect lexicons, sentiment lexicons, and word embeddings from AraVec and fastText. AraVec embeddings outperformed the other features. Mulki et al. [31] also used SVM with TF–IDF to represent features, their main approach was in testing different preprocessing steps. Abdullah and Shaikh [32] used deep learning techniques by utilizing AraVec word embeddings and feeding them into four dense neural networks (DNNs).

In [33], a deep learning system called binary neural network was proposed and evaluated on the English dataset of SemEval 2018 task 1 [19]. The system used three embedding models coupled with an attention function. It achieved better performance on the multi-label emotion classification subtask than the systems that participated in the task. However in [34], sentiment and emotion aware word embeddings were constructed by combining semantic word vectors and emotion word vectors to generate hybrid sentiment-aware word representations. These proposed embeddings were used for emotion classification of a Weibo dataset of eight classes (happiness, trust, anger, sadness, fear, disgust, none). Several classifiers were evaluated (SVM, logistic regression, decision tree, gradient boost). The gradient boost model reported the best F1-score when used with the hybrid sentiment-aware embeddings. In [35], a new feature selection scheme for emotion classification was proposed and compared to other feature reduction techniques, namely chi-square, Gini-text, and delta. The proposed scheme was called relevance score and was proved to improve the classification of emotions.

As for the work on Arabic emotion analysis, earlier studies used either unsupervised lexicon based approaches such as [36] or supervised machine learning approaches such as [20,37,38] that use SVM and multinomial naive Bayes (MNB). Recently the shift has been to deep learning models such as in [39], where a CNN-LSTM was used to detect both emotions and sentiment in Arabic tweets. In [40], AraNet, a deep learning toolkit for Arabic social media processing was proposed. AraNet predicts age, dialect, gender, emotion, irony, and sentiment from social media posts. It utilizes the recent multilingual BERT model.

One of the early works on developing datasets for Arabic emotions was presented in [21], where a multi-dialect data set for Arabic emotion analysis called DINA was curated. The collection of the dataset depends on a seed list of Arabic emotions based on the Ekman classification of six emotions. The Twitter API was polled using this seed list. Then for each of the six emotions, 500 tweets were selected resulting in a dataset of 3000 tweets. The dataset was manually annotated by human annotators. For each of the six emotions the annotators were asked to determine the respective emotion first, then the emotion intensity according to the set {zero, weak, fair, strong}. A thorough manual analysis of the dataset and the annotations was also presented.

Alhuzali et al. in [22] also developed a new dataset of Arabic tweets for emotion detection. Their approach is based on using a list of Arabic seeds for each of the Plutchik primary emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, trust. Then two datasets were composed. The first dataset called LAMA is composed of 8000 tweets (1000 for each of the eight emotions) that was manually annotated by four annotators. After cleaning and removing duplicates, this dataset consisted of 7268 tweets. The second dataset called LAMA-DIST was composed through distant-supervision by using the same seed list of eight emotions. After cleaning and removing duplicates and tweets less than five words, the resulting dataset comprises 182,690 tweets. Then a gated recurrent neural network was used for emotion classification, which was compared to a baseline that uses SVM. The proposed methods (supervised, distant supervised, and hybrid supervised models) delivered comparable performance on both datasets.

Another attempt to provide a dataset of Arabic tweets annotated for emotions can be found in [20], where a dataset of 10,000 tweets were annotated for eight emotions: joy, happiness, anger, sympathy, sadness, fear, surprise, love, none. The dataset was manually annotated by human annotators. Preliminary experiments were performed for emotion detection by utilizing naive Bayes and SVM.

However, in [37], the authors proposed an automatic approach for emotion annotation that relies on emojis. A comparison with a manually annotated dataset had proven the feasibility of the proposed approach that was evaluated using two classifiers: SVM and MNB. The results of the tests show that the automatic labeling approaches using SVM and MNB outperform manual labeling approaches.

In [41], three models were presented for Arabic emotion recognition: deep feature-based model (DF), human engineered feature-based model (HEF), and the combination of them referred to as (HEF + DF) a hybrid model. The performance of the proposed models were evaluated on the SemEval 2018 [19], Iraqi Arabic Emotion Dataset (IAEDS) [42], and Arabic Emotions Twitter Dataset (AETD) [20] datasets. The HEF model used a set of semantic, syntactic, and lexical human engineered features. While the DF model used a combination of embedding layers: Emoji2vec, AraVec, GloVeEmb, and FastTextEmb. The best performing model on all the datasets is the hybrid (HEF + DF) model.

After reviewing the related work on Arabic emotion analysis, we observe that most of the previous work used either traditional machine learning with features such as TF–IDF and/or emotion lexicons, or deep learning approaches that utilize non-contextual embeddings such as word2vec or fastText. However, we have not found any previous work that has attempted to use PLMs and transfer learning for the task of Arabic emotion analysis.

3. Materials and Methods

3.1. Dataset

As mentioned in the previous section, public datasets for emotion analysis in Arabic are limited. In this paper, we used the SemEval 2018 [19] emotion dataset. The data is a collection of Arabic tweets that were annotated for four basic emotions: anger, sadness, joy, fear. We ignored the emotion intensity since this is not our main objective in the study and removed tweets that were annotated as not holding any emotion. The resulting number of tweets and their associated emotion are shown in Table 2. We used a train/test split of 80/20 for the first study. Then we added additional experiments in the second study using k-fold cross validation with k = 5 to minimize dataset split bias.

Table 2. Dataset size, labels, and train/test split.

Emotion	Train	Test	Total
anger	827	210	1037
fear	715	181	896
joy	953	237	1190
sadness	674	165	839
Total	3169	793	3962

3.2. Modeling Study

All experiments were performed using Google Colab with GPU, the huggingface transformers library (https://huggingface.co/, accessed on 31 December 2020), and the new FastBert library (https://github.com/kaushaltrivedi/fast-bert, accessed on 31 December 2020) that was specifically designed for multi-class text classification tasks.

3.2.1. First Study

Since the main objective in this paper is to study the impact of the evolution of language models, the experiments conducted in the first study are as follows:

1. Emotion classification using SVM and TF–IDF as features (one experiment).
2. Emotion classification using SVM and different versions of word2vec as features (four experiments).

3. Emotion classification by finetuning on different monolingual BERT models for Arabic (five experiments).

In total, 10 different experiments were conducted. For the word2vec model, we used the AraVec [26] which is a set of word2vec models for the Arabic language that was presented in Section 2 of this paper. Specifically, we used the model that was trained on a large Twitter corpus. We used the n-grams model, this model has four versions: the continuous bag of words (CBOW) model and the skip-gram (SG) model—each model has two versions according to dimension size: 100 and 300.

As for the BERT models for Arabic, we used the two versions of AraBERT [23], the base and large versions of ArabicBERT [24] and the multi-dialect-ArabicBERT [25] in the experiments (all of these models were presented in Section 2). All of the experiments were done using an Adam optimizer with a learning rate of 2×10^{-6} and a batch size of 8 for 10 epochs. The max sequence length was set to 256. The finetuning was performed by adding a softmax classification layer to the model that performs the emotion classification.

To handle data preprocessing, we used regular expressions to clean the data from noise, i.e., removing numbers, English characters, links, hashtags, user mentions, and punctuation, as well as normalizing Arabic characters.

For evaluation, we report the F1-score for each emotion *e*, and the macro and micro F1 score for all emotion classes. Let $\mathrm{P_e}$, $\mathrm{R_e}$, and $\mathrm{F1_e}$ denote the precision score, recall score, and the F1-score of the emotion e.

$$\mathrm{P_e} = tp/(tp + fp) \tag{1}$$

$$\mathrm{R_e} = tp/(tp + fn) \tag{2}$$

$$\mathrm{F1_e} = 2\mathrm{P_e}\mathrm{R_e}/(\mathrm{P_e} + \mathrm{R_e}) \tag{3}$$

where for each emotion label e:

- *tp* is the number of tweets classified correctly to the correct emotion *e* (true positive),
- *fp* is the number of tweets falsely classified to the emotion *e* (false positive),
- and *fn* is the number of tweets that should have been classified as emotion *e* but were falsely classified to another emotion (false negative).

The macro-averaged F1-score is calculated as follows:

$$\mathrm{F1_{macro}} \frac{1}{|\mathrm{e}|} \sum \mathrm{F1_e} \tag{4}$$

The micro-averaged F1-score is calculated as follows:

$$\mathrm{F1_{micro}} = 2\mathrm{P_{micro}}\mathrm{R_{micro}}/(\mathrm{P_{micro}} + \mathrm{R_{micro}}) \tag{5}$$

We also show the confusion matrix and precision-recall curve for each experiment to analyze the classification of the different classes.

3.2.2. Second Study

In the second study the same experiments done in the first study for the BERT models were also conducted but using k-fold cross validation.

4. Results and Discussion

The results for all 10 experiments in first study are shown in Table 3.

From the results in Table 3, we can observe how the performance increases as we adopt new LMs. First, we notice the significant increase in performance (+11–14%) from the TF–IDF model compared to the AraVec (word2vec) models. This was surely expected, due to the superiority of the word2vec model compared to the traditional TF–IDF. Moreover, comparing the different versions of the AraVec models, we can see that the SG models outperform the CBOW models even when the dimensions are different. It is important to note that we are comparing and reporting on the macro-average F1-score since the classification problem we are working on is multi-class and the performance on all the classes is equally important.

Table 3. Results for all experiments in first study: F1-score for each emotion, macro F1, and micro F1.

Method	Anger-F1	Fear-F1	Sad-F1	Joy-F1	Macro F1	Micro F1
TF-IDF	0.47	0.48	0.41	0.81	0.54	0.56
AraVecCBOW100	0.60	0.52	0.57	0.91	0.65	0.67
AraVecSG100	0.62	0.53	0.56	0.93	0.66	0.68
AraVecCBOW300	0.60	0.53	0.52	0.87	0.63	0.65
AraVecSG300	0.64	0.58	0.57	0.92	0.68	0.70
AraBertv01	0.66	0.64	0.68	0.92	0.73	0.74
AraBertv1	0.67	0.65	0.68	0.91	0.73	0.74
ArabicBertBase	0.66	0.60	0.66	0.92	0.71	0.73
ArabicBertLarge	**0.69**	**0.67**	**0.71**	**0.95**	**0.76**	**0.77**
Multi-DialectBert	0.64	0.64	0.68	0.92	0.72	0.73

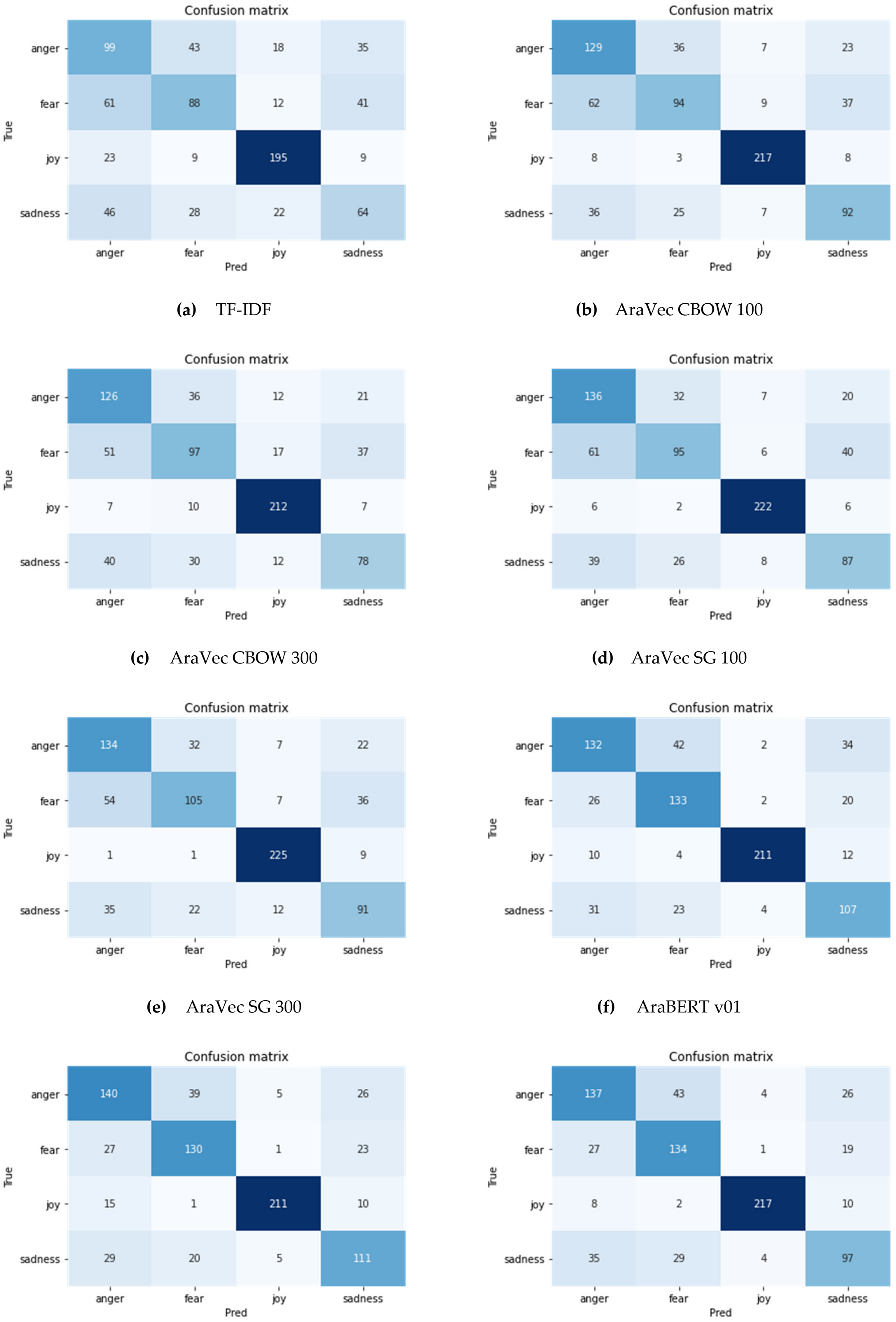
As for the BERT models, we observe the significant increase when compared to TF-IDF (+19–22%) and also when compared to word2vec models (+10–13%). This truly shows the dominance of the PLMs especially given the small size of the dataset; hence, transfer learning is a promising solution for the cases where the labeled data is limited.

Analysis of different BERT models for Arabic: There was no difference in performance between the two versions of AraBERT with and without Farasa segmentation. The reason could be that the Farasa tokenizer does not perform well on dialectal and noisy data of the Twitter genre since it was trained on modern standard Arabic (MSA) [27] which is the formal form of Arabic.

As for the AraBERT and ArabicBERT models, we notice that AraBERT and ArabicBERT-Base were very close in performance where AraBERT was (+2%) better, the reason could be that these two models share the same architectures in terms of hidden layer size, attention heads, and number of parameters as we saw in Table 2. However, the best performance from all the models was achieved by ArabicBERT large, this is not a surprise, since this model has a large number of parameters as shown in Table 2 also.

Moreover, when comparing the different versions of ArabicBERT to the multi-dialect-ArabicBERT. It was expected that the multi-dialect-ArabicBert perform better than the original ArabicBERT-Base, but we found that they gave similar performance. The tweets dataset used for additional pre-training in the multi-dialect-ArabicBert was small compared to the original dataset. Moreover, given that one of the objectives of BERT is next sentence prediction, which cannot be applied to tweets since they are mostly composed of one sentence and not long paragraphs, this could be another reason that this version did not give any further improvement.

We observe from the confusion matrices of all models in Figure 2, that there was a very low confusion for the joy emotion with other emotions; the reasons could be that the joy emotion has the largest number of instances in the dataset and that it stands out as a positive emotion which is represented differently than negative emotions. Also, although the dataset size is considered small and limited, we still observed a high recall for the positive emotion joy as shown in Figure 3, proving that these language models have a high capability to distinguish between positive and negative emotions. Moreover, the highest confusion was between the anger and fear emotions. We also notice from Figure 3 and the F1-score for each emotion in Table 3, that the AraVec models were able to classify the anger emotion correctly among the negative emotions, while the BERT models were able to classify the sadness emotion among the negative emotions. The reason could be the out-of-vocabulary problem that word2vec suffers from.



(a) TF-IDF

(b) AraVec CBOW 100

(c) AraVec CBOW 300

(d) AraVec SG 100

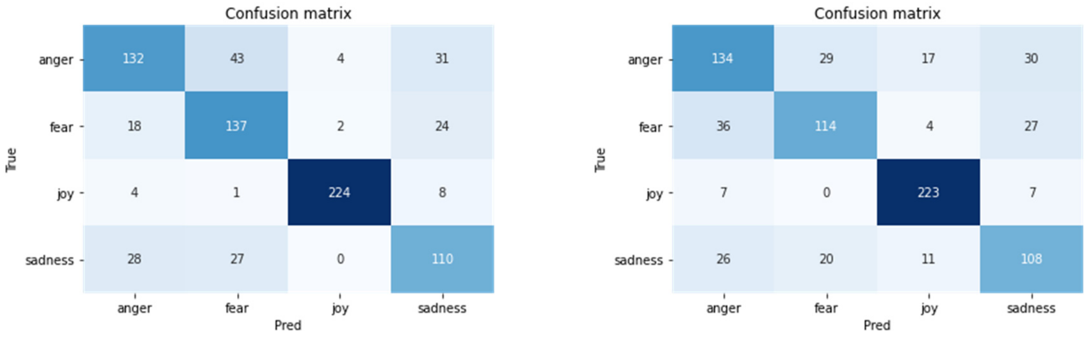
(e) AraVec SG 300

(f) AraBERT v01

(g) AraBERT v1

(h) ArabicBERT-Base

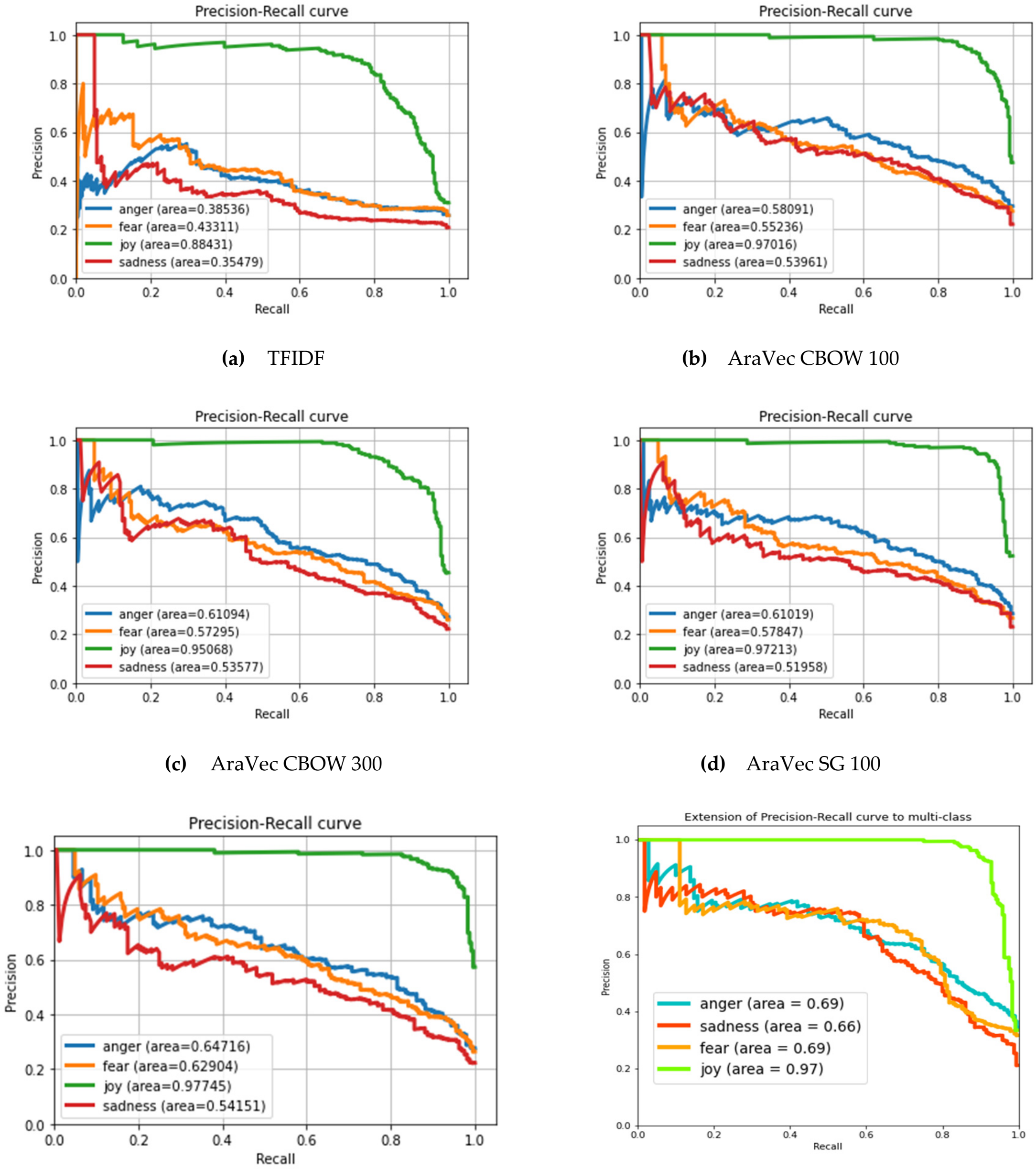
Figure 2. *Cont.*



(i) ArabicBERT-Large

(j) MultiDialect ArabicBERT

Figure 2. The confusion matrix for each of the 10 experiments in first study (**a**–**j**) for each language model.



(a) TFIDF

(b) AraVec CBOW 100

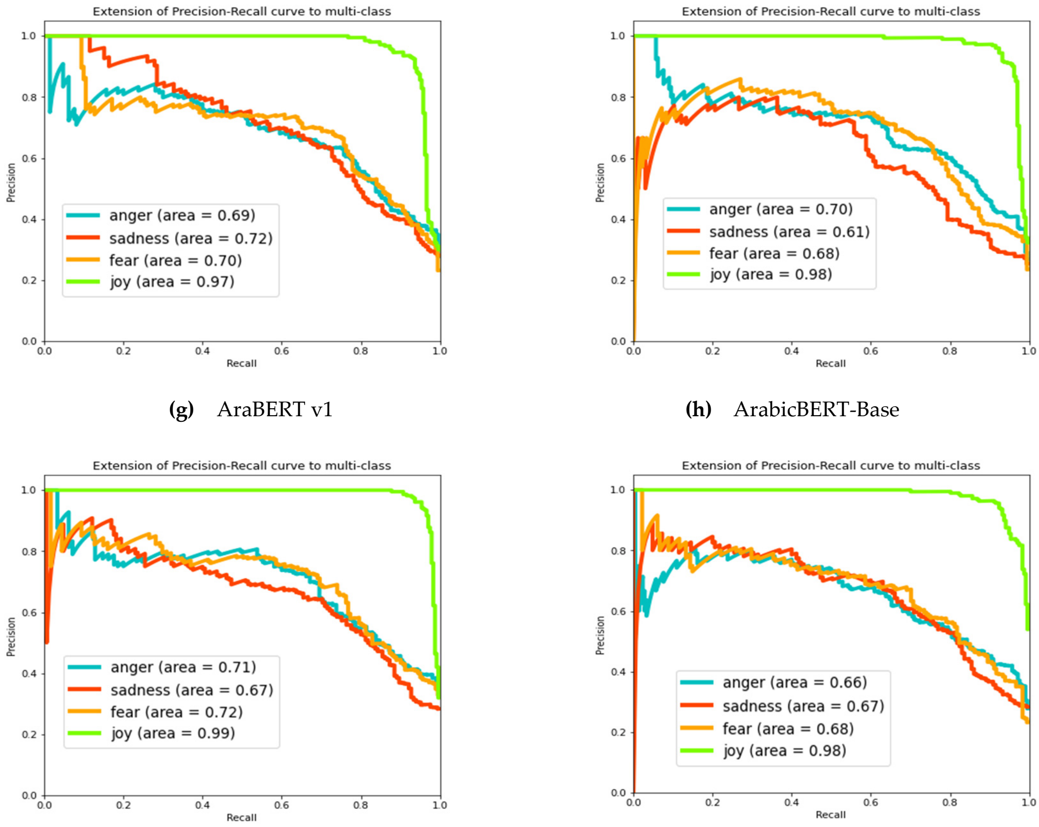
(c) AraVec CBOW 300

(d) AraVec SG 100

(e) AraVec SG 300

(f) AraBERT v01

Figure 3. *Cont.*



(g) AraBERT v1

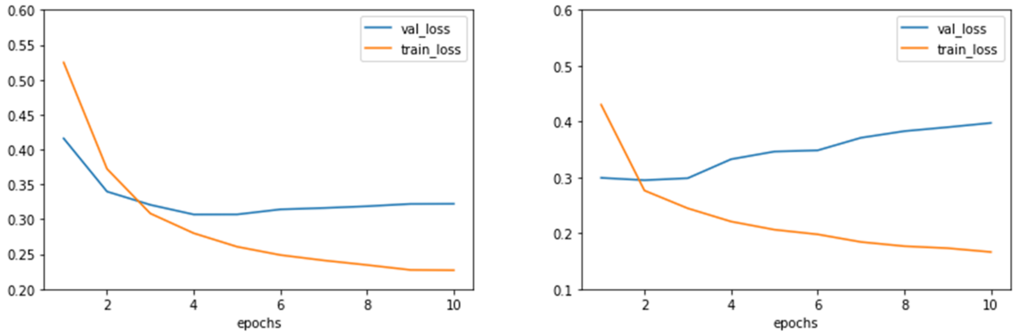
(h) ArabicBERT-Base

(i) ArabicBERT-Large

(j) MultiDialect ArabicBERT

Figure 3. The precision-recall curve for all 10 experiments in first study (**a**–**j**) for each language model.

To assess the learning capacity of the BERT models used in this study we show in Figure 4 the epochs vs. loss learning curves for the AraBERT v1 and the ArabicBERT-Large models. We can see that the AraBERTv1 reaches convergence while the ArabicBERT-Large model suffers from overfitting. To resolve the overfitting in ArabicBERT-Large we perform early stopping (epochs = 3); the macro F1 and micro F1 become 0.75 and 0.76, respectively, hence only a decrease of 1% when compared to the results reported in Table 3.



(a) AraBERT v1

(b) ArabicBERT-Large

Figure 4. The learning curve for (**a**) AraBERT v1 and (**b**) ArabicBERT-Large.

Limitations of study: Although the results of the empirical study that was conducted in this paper show the promising performance results of PLMs, these models suffer from long training times. The training times in minutes for each model respectively were: AraBERT v01: 43, AraBERT v1: 32, ArabicBERT-Base: 32, ArabicBERT-Large: 105, MultiDialect ArabicBERT: 32, while the AraVec models took around 7 min. Moreover, we suggest for future work to add more emotion classes to the dataset in addition to text that does not contain any emotion.

To minimize dataset split bias, we repeated the experiments for the BERT models but using cross validation with k = 5, the results of this second study are shown in Table 4. We observe the slight decrease in macro and micro F1-score for all the models, nonetheless they still give higher performance than the traditional language models. We also report the 95% confidence interval for cross validation as shown in Table 4.

Table 4. Results for second study: five-fold cross validation.

Method	Macro F1	Micro F1	Confidence Interval
AraBertv01	0.73	0.74	74 (+/− 0.03)
AraBertv1	0.72	0.73	73 (+/− 0.02)
ArabicBertBase	0.70	0.72	72 (+/− 0.04)
ArabicBertLarge	0.72	0.74	74 (+/− 0.04)
Multi-DialectBert	0.70	0.71	71 (+/− 0.02)

5. Conclusions

In this paper, we presented an empirical study on the evolution of language models from the traditional TF–IDF to the more sophisticated word embedding word2vec and finally the recent state-of-the-art pretrained language model BERT. We also investigated different BERT models for Arabic by applying the experiments on a downstream task that has not received much attention from the research community: emotion analysis.

We find that the best performance is achieved with the ArabicBERT-Large model, which is a BERT model trained on a large dataset of Arabic text. The increase in F1-score was significant at +7–21%. The dataset we performed the experiments on was relatively small. This leads to the conclusion that there is no need for the intensive annotation of large datasets that are required for deep learning techniques or traditional ML as PLMs can be fine-tuned on relatively small, annotated task-specific datasets.

For future work we suggest to pre-train new PLMs other than BERT such as ROBERTa or XLNet for the Arabic language. We also argue that different genres other than Wikipedia or news articles should be considered for the pretraining such as large datasets of Arabic tweets.

Funding: This research project was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges," Deanship of Scientific Research, King Saud University.

Data Availability Statement: Data is contained within the article. The data presented in this study are available in [19,23,24].

Conflicts of Interest: The author declares no conflict of interest.

References

1. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the Advances in Neural Information Processing Systems 26, Lake Tahoe, NV, USA, 5–10 December 2013; p. 9.
2. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
3. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
4. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
5. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.

6. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training (2018). Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 31 December 2020).
7. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692[cs]2019.
8. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 5753–5763.
9. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. *arXiv* **2020**, arXiv:1909.11942 [cs] 2020.
10. Clark, K.; Luong, M.-T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators. In Proceedings of the ICLR, Addis Ababa, Ethiopia, 26–30 April 2020.
11. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-Trained Models for Natural Language Processing: A Survey. *arXiv* **2020**, arXiv:2003.08271 [cs] 2020. [CrossRef]
12. Mohammad, S.; Kiritchenko, S. Understanding Emotions: A Dataset of Tweets to Study Interactions between Affect Categories. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
13. Ekman, P. An Argument for Basic Emotions. *Cogn. Emot.* **1992**, *6*, 169–200. [CrossRef]
14. Plutchik, R. A general psychoevolutionary theory of emotion. In *Theories of Emotion*; Elsevier: San Leandro, CA, USA, 1980; pp. 3–33.
15. Cherry, C.; Mohammad, S.M.; Bruijn, B.D. Binary Classifiers and Latent Sequence Models for Emotion Detection in Suicide Notes. *Biomed. Inform. Insights* 2012. [CrossRef] [PubMed]
16. Jabreel, M.; Moreno, A.; Huertas, A. Do local residents and visitors express the same sentiments on destinations through social media? In *Information and Communication Technologies in Tourism*; Springer: Cham, Switzerland, 2017; pp. 655–668.
17. Mohammad, S.M.; Zhu, X.; Kiritchenko, S.; Martin, J. Sentiment, Emotion, Purpose, and Style in Electoral Tweets. *Inf. Process. Manag.* **2015**, *51*, 480–499, . [CrossRef]
18. Cambria, E. Affective Computing and Sentiment Analysis. *IEEE Intell. Syst.* **2016**, *31*, 102–107. [CrossRef]
19. Mohammad, S.; Bravo-Marquez, F.; Salameh, M.; Kiritchenko, S. SemEval-2018 Task 1: Affect in Tweets. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 23 April 2018; pp. 1–17.
20. Al-Khatib, A.; El-Beltagy, S.R. Emotional Tone Detection in Arabic Tweets. In Proceedings of the Computational Linguistics and Intelligent Text Processing, Hanoi, Vietnam, 18–24 March 2018; pp. 105–114.
21. Abdul-Mageed, M.; AlHuzli, H.; Abu Elhija, D.; Diab, M. DINA: A Multi-Dialect Dataset for Arabic Emotion Analysis. In Proceedings of the 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media held in conjunction with the 10th International Conference on Language Resources and Evaluation (LREC2016), Portorož, Slovenia, 23–28 May 2016.
22. Alhuzali, H.; Abdul-Mageed, M.; Ungar, L. Enabling Deep Learning of Emotion with First-Person Seed Expressions. In Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, New Orleans, LA, USA, 5–6 June 2018; pp. 25–35.
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; pp. 9–15.
24. Safaya, A.; Abdullatif, M.; Yuret, D. KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. In Proceedings of the Fourteenth Workshop on Semantic Evaluation; International Committee for Computational Linguistics: Barcelona (online), Barcelona, Spain, 12–13 December 2020; pp. 2054–2059.
25. Talafha, B.; Ali, M.; Za'ter, M.E.; Seelawi, H.; Tuffaha, I.; Samir, M.; Farhan, W.; Al-Natsheh, H.T. Multi-Dialect Arabic BERT for Country-Level Dialect Identification. In Proceedings of the Fifth Arabic Natural Language Processing Workshop (WANLP2020), Barcelona, Spain, 12 December 2020.
26. Soliman, A.B.; Eissa, K.; El-Beltagy, S.R. AraVec: A Set of Arabic Word Embedding Models for Use in Arabic NLP. *Procedia Comput. Sci.* **2017**, *117*, 256–265, . [CrossRef]
27. Abdelali, A.; Darwish, K.; Durrani, N.; Mubarak, H. Farasa: A Fast and Furious Segmenter for Arabic. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, San Diego, CA, USA, 12–17 June 2016; pp. 11–16.
28. Zampieri, M.; Nakov, P.; Rosenthal, S.; Atanasova, P.; Karadzhov, G.; Mubarak, H.; Derczynski, L.; Pitenis, Z.; Çöltekin, Ç. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In Proceedings of the Fourteenth Workshop on Semantic Evaluation; International Committee for Computational Linguistics, Barcelona, Spain, 12–13 December 2020; pp. 1425–1447.
29. Abdul-Mageed, M.; Zhang, C.; Bouamor, H.; Habash, N. NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task. In Proceedings of the Fifth Arabic Natural Language Processing Workshop; Association for Computational Linguistics, Barcelona, Spain, 12 December 2020; pp. 97–110.

30. Badaro, G.; El Jundi, O.; Khaddaj, A.; Maarouf, A.; Kain, R.; Hajj, H.; El-Hajj, W. EMA at SemEval-2018 Task 1: Emotion Mining for Arabic. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 236–244.
31. Mulki, H.; Bechikh Ali, C.; Haddad, H.; Babaoğlu, I. Tw-StAR at SemEval-2018 Task 1: Preprocessing Impact on Multi-Label Emotion Classification. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 167–171.
32. Abdullah, M.; Shaikh, S. 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, New Orleans, LA, USA, 5–6 June 2018; pp. 350–357.
33. Jabreel, M.; Moreno, A. A Deep Learning-Based Approach for Multi-Label Emotion Classification in Tweets. *Appl. Sci.* **2019**, *9*, 1123. [CrossRef]
34. Mao, X.; Chang, S.; Shi, J.; Li, F.; Shi, R. Sentiment-Aware Word Embedding for Emotion Classification. *Appl. Sci.* **2019**, *9*, 1334. [CrossRef]
35. Erenel, Z.; Adegboye, O.R.; Kusetogullari, H. A New Feature Selection Scheme for Emotion Recognition from Text. *Appl. Sci.* **2020**, *10*, 5351. [CrossRef]
36. Al-A'abed, M.; Al-Ayyoub, M. A Lexicon-Based Approach for Emotion Analysis of Arabic Social Media Content. In Proceedings of the The International Computer Sciences and Informatics Conference (ICSIC), Amman, Jordan, 12–13 January 2016.
37. Hussien, W.A.; Tashtoush, Y.M.; Al-Ayyoub, M.; Al-Kabi, M.N. Are Emoticons Good Enough to Train Emotion Classifiers of Arabic Tweets? In Proceedings of the 2016 7th International Conference on Computer Science and Information Technology (csit), Amman, Jordan, 13–14 July 2016; IEEE: New York, NY, USA, 2016. ISBN 978-1-4673-8913-6.
38. 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), Kuala Lumpur, Malaysia, 17–19 November 2014; pp. 37–49.
39. 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.
40. Abdul-Mageed, M.; Zhang, C.; Hashemi, A.; Nagoudi, E.M.B. AraNet: A Deep Learning Toolkit for Arabic Social Media. 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; pp. 16–23.
41. Alswaidan, N.; Menai, M.E.B. Hybrid Feature Model for Emotion Recognition in Arabic Text. *IEEE Access* **2020**, *8*, 37843–37854, . [CrossRef]
42. Almahdawi, A.J.; Teahan, W.J. A New Arabic Dataset for Emotion Recognition. In *Proceedings of the Intelligent Computing*; Arai, K., Bhatia, R., Kapoor, S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 200–216.