Applying Recurrent Networks For Arabic Sentiment Analysis

Eslam Omara
Information Center
Agriculture Directorate
Cairo Governorate
Cairo, Egypt
e_omara@hotmail.com

Mervat Mosa
Computer Science and Engineering
Faculty of Electronic Engineering
Menofia University
Menouf, Egypt
mervat_mosa@yahoo.com

Nabil Ismail
Computer Science and Engineering
Faculty of Electronic Engineering
Menofia University
Menouf, Egypt
nabil_a_ismail@yahoo.com

***Abstract*—The main characteristic of deep learning approaches is the ability to learn differentiating and discriminating features. These techniques can discover complex relations and structures within high-dimensional data. For feature extraction, deep learning models employ several layers of nonlinear processing units. One of the fields that have applied deep architectures with a noticeable breakthrough in performance measures is Natural Language Processing (NLP). Recurrent neural networks (RNNs) and their variants Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are commonly used for NLP applications as they are efficient at processing sequential data. Unlike RNNs, LSTMs and GRUs can combat vanishing and exploding gradients. In Addition, Convolutional Neural Network (CNN) is another deep architecture that has been widely used in language processing. On the other side, sentiment analysis (SA) is an NLP task concerned with opinions, attitudes, emotions, and feelings. Sentiment analysis deduces the author's attitude regarding a topic and classifies the attitude polarity according to a set of predefined classes. Application of SA in business analytics helps to gain insight into consumer behaviour and needs. In the proposed work deep LSTM, GRU, and CNN are applied for Arabic sentiment analysis. The models are implemented and tested employing character-level representation. Also, deep hybrid models that combine multiple layers of CNN with LSTM or GRU are studied. The application aims at investigating the capability of deep LSTM, GRU, and hybrid architectures to learn and extract features from character-level representation. Results show that combining different architectures can boost performance in SA tasks. The CNN-LSTM/GRU combinations registered higher accuracy compared to deep LSTM and GRU.**

Keywords— Deep learning; Sentiment analysis; LSTM; GRU; CNN-LSTM; CNN-GRU.

I. INTRODUCTION

NLP considers many tasks that aim at analyzing text structures and understanding text semantics. The extracted syntactic and semantic information is then exploited for a higher level target. Examples of NLP tasks are Named entity recognition (NER) [1], Part-of-speech Tagging (POS) [1], Chunking or shallow parsing [2], Parsing [3], Word-sense disambiguation [4], Anaphora resolution (pronoun resolution) [5], Semantic role labeling (SRL) [1], Sentence classification [6], Sentiment analysis [7], Emotion detection (ED) [8, 9], Document classification [10], Text summarization [11], Machine translation [3], and Question answering (QA) [2].

Recently, deep architectures have been extensively applied in NLP. Models that employ deep structures to identify and extract relevant features from large data corpora have reported enhanced performance in many fields [12]. In addition to NLP, deep structures have been employed in various fields as computer vision, handwriting recognition, speech recognition, object detection, cancer detection, biological image classification, face recognition, stock market analysis, and others [13].

RNNs are commonly used for sequence modelling. The recurrence connection enables memorizing information as the context in natural language tasks [14]. RNNs are widely implemented in NLP as they can consider the word order which enables preserving the context [15]. Unlike feedforward neural networks that use the learned weights for output prediction, RNN makes use of the learned weights and a state vector for output generation [16]. RNNs have two variants Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). The two variants are based on the notion of gates [16, 17]. On contrary to RNN, gated variants are capable of handling long-term dependencies. Also, they can combat vanishing and exploding gradients by the gating technique [14]. LSTM is the most widespread deep architecture applied to NLP as it has shown the ability to capture far distance dependency of terms [15]. Also, GRUs have been investigated in many NLP tasks as they can train faster and suit smaller datasets compared to LSTM [17]. The other popular structure that has been applied for sentiment analysis is CNNs. Deep and shallow CNNs have been studied at word and character level representation [18]. Moreover, hybrid architectures have shown the capability to abstract task-related features, consider word order, and extract long context information in sentiment analysis [19]. Convolutional layers extract more abstracted semantic features from text and reduce the dimensionality and RNN layers capture the context.

On the other hand, many customers create and share content about their experience on review sites, social channels, and blogs. Writers' tweets, reviews, comments, and form submissions, involve information that could prove useful for making informed decisions. Sentiment analysis aims at studying and understanding people's emotions, behaviours, opinions, feelings, and assessments regarding different targets as services, facilities, products, problems, items, firms, occasions, topics, and even people as public figures [8]. By applying NLP techniques, SA

detects the polarity of the opinionated text and categorizes it as positive, negative, or neutral.

The proposed work objective is to study the application of deep LSTMs, GRUs, and hybrid architectures based on character level representation to boost SA performance in the Arabic Language. The main contributions of the proposed paper are :

- Investigation of deep LSTM and GRU models for Arabic sentiment classification based on character level features.
- Applying deep hybrid CNN-LSTM and CNN-GRU models that merge layers of different architectures for Arabic SA.
- Exploring the ability of deep networks to extract discriminating features from data represented at the character level.
- The performance of the models is experimented with and compared to find the model that best fits the low-level representation.

The paper is organized as follows: Section II clarifies the sentiment analysis task and section III discusses NLP feature representation. CNN, LSTM, and GRU are outlined in section IV. Related work is introduced in section V. In section VI the applied network structure and settings are explained. Experimental results are proposed in section VII. Conclusion and further future work are declared in sectionVIII.

II. SENTIMENT ANALYSIS

SA research depends on data repositories that include tweets, reviews, and comments. Topics discussed recently are mobile devices, the stock market, and human emotions while early topics include reviews, product features, and elections [20]. Mining sentiment has been studied at multiple granularity levels, namely document level, sentence level, and aspect level. At a document level, each opinionated text is considered as one unit and assigned a positive, negative, or neutral polarity. At this level, a document holds an opinion regarding a single entity and the document has one opinion holder. Documents that maintain multiple entities assessment cannot be analyzed using this level [6, 21]. Sentence level SA begins with determining if the sentence expresses an opinion or not (subjective or objective) this step is known as subjectivity classification. And then the sentiment orientation of subjective sentences is identified by multi-class or binary classification. Multi-class classification assigns a category positive, negative, or neutral to the sentence while binary categorize sentences as positive or negative [6, 21, 22].

A more fine-grained SA is the aspect level or phrase level that defines the quintuple (O; A; SO; H; T) of an opinion about an entity or an entity feature. It is also called feature-based sentiment analysis. The main tasks for this level involve aspect extraction and aspect sentiment classification. In aspect extraction, the entities or entity attributes are detected. In aspect sentiment classification, the author's orientation towards the aspect is determined. An opinion about an object may hold a positive orientation for an aspect and a negative orientation for another aspect, so it is not positive or negative for the whole object [6, 21, 22].

III. FEATURE REPRESENTATION

Text cannot be processed in its raw format and so it has to be transformed into a standard representation. Selecting the appropriate representation that most suits the application is an essential step [23]. The common methodologies used to represent text as vectors are Vector Space Model (VSM) and neural network-based representation. The vector can be a representation of a character, word, paragraph, or whole document. An early applied approach is to represent text documents using binary representation. The representation is a very large sparse matrix resulting in a high dimensionality problem [23]. Representation vectors are built with one-hot encoding where no information about word meaning is preserved. One-hot encoding does not distinguish similar words from completely different worlds. This representation is referred to as discrete representation or local representation [24].

The bag of Word (BOW) approach constructs a vector representation of a document based on the term frequency in the corpora [23, 24]. BOW is widely speared for text classification applications. However, a drawback of BOW representation is that word order is not preserved, leading to the loss of the semantic relation between words. Another limitation is that each word is represented as a distinct dimension. And hence, the representation vectors are sparse with too many dimensions equal to the corpus vocabulary size.

The multi-word term is another approach where vectors encode terms composed of multiple words. This needs a complicated algorithm to extract terms from documents [23]. N-Grams scheme uses sequences of words extracted from documents as features. But, the number of words selected for effectively representing documents is difficult to determine [23]. Bag-Of-N-Grams (BONG) is one of the variations of BOW. The representation vocabulary is extended by appending a set of consecutive words to the word set. The main drawback of BONG is more sparsity and higher dimensionality compared to BOW [24]. Bag-Of-Concepts is also a document representation approach where every dimension is related to a general concept described by one or multiple words [24].

Alternatively, words can be encoded by a continuously distributed representation. Each word is assigned a continuous vector that belongs to a low-dimensional vector space. Neural networks are commonly used for learning a distributed representation of text known as word embedding [24]. Popular neural models used for learning word embedding are Continuous Bag-Of-Words (CBOW), Skip-Gram, and GloVe embedding. A discriminant feature of word embedding is that they are powerful at capturing semantic and syntactic connections among words. Embedding vectors of semantically similar or syntactically similar words are close vectors with high similarity.

IV. DEEP ARCHITECTURES

Deep learning models differ from Machine Learning (ML) in how to extract features. Traditional machine learning techniques apply handcrafted features using extraction approaches, then train the learning algorithm.

Deep models employ layers of deep architectures to learn features and represent them in multiple hierarchical levels [25]. Furthermore, they can compose an inner state space by tracing the seen inputs and building a memory using the recurrence approach [26].

A. Convolutional Neural Network (CNN)

CNN preserves spatial locality in the input text and learns multiple levels of abstracted representation. With one layer, simple patterns are learned, and stacking many layers enables multiple pattern extraction [16]. The applied CNN is structured of one dimension convolution and one dimension max-pooling [27]: First: Temporal convolutional modules compute a 1-D convolution for a discrete input function $g(x) \in [1, l] \rightarrow \mathbb{R}$ and a discrete kernel function $(x) \in [1, k] \rightarrow \mathbb{R}$. The convolution $h(y) \in [1, \lfloor (l-k)/d \rfloor + 1] \rightarrow \mathbb{R}$ Between $f(x)$ and $g(x)$ with stride, d is defined as:

$$h(y) = \sum_{x=1}^{k} f(x) \, . \, g(y \, . \, d - x + c) \quad (1)$$

Where $c = k - d + 1$ is an offset constant. The module is parameterized by a set of kernel functions $f_{ij}(x)$ for $i = 1, 2, \ldots m$ and $j = 1, 2, \ldots n$ which represents weights on a set of inputs $g_i(x)$ and outputs $h_j(y)$. g_i refers to the input features, and m is the input feature size. h_j refers to the output features, and n is the output feature size. The output $h_j(y)$ is obtained by a sum over i of the convolutions between $g_i(x)$ and $f_{ij}(x)$.

Second: Temporal max-pooling is the 1-D version of the max-pooling module which enables the training of deep models with more than 6 layers for a discrete input function $g(x) \in [1, l] \rightarrow \mathbb{R}$. The max-pooling function $h(y) \in [1, \lfloor (l-k)/d \rfloor + 1] \rightarrow \mathbb{R}$ Is defined as:

$$h(y) = \max_{k \geq x \geq 1} g(y \, . \, d - x + c) \quad (2)$$

Where $c = k - d + 1$ is an offset constant. Convolution is conducted by sliding the kernel along with the input signal which is referred to as shift-compute [25].

B. Long Short Term Memory (LSTM)

LSTM is a variant of simple RNN which can learn long scale dependencies [16, 17]. LSTM is the most applied type of RNN, and they are appropriate for processing temporal data [25]. LSTMs employ input gate, forget gate, output gate, internal state, and the cell state to manipulate the vanishing gradient problem [16]. The memory cell remembers values all over the time dimension. The gates are responsible for controlling the information flow to and out of the cell [28]. The parameters of the gates are learned during training. To calculate the hidden state h_t from the prior hidden state h_{t-1} the next equations are calculated [16, 17]:

$$i_t = \sigma(U_i \, x_t + W_i \, h_{t-1}) \quad (3)$$

$$f_t = \sigma(U_f \, x_t + W_f \, h_{t-1}) \quad (4)$$

$$o_t = \sigma(U_o x_t + W_o \, h_{t-1}) \quad (5)$$

$$g_t = tanh(U_g \, x_t + W_g \, h_{t-1}) \quad (6)$$

$$c_t = (c_{t-1} \otimes f_t) \oplus (g_t \otimes i_t) \quad (7)$$

$$h_t = tanh(c_t) \otimes o_t \quad (8)$$

Where i the input gate, f the forget gate, and o the output gate. The cell state c is the interior memory of the cell.

C. Gated Recurrent Unit(GRU)

GRU is a variation of LSTM that can handle vanishing gradient problems. GRUs are more simplified regarding topology, computational cost, and complexity [25], and hence, they train faster compared to LSTM [16]. The inner architecture is less complicated and fewer calculations are required to update the hidden state. The subsequent equations illustrate the gating methodology in the GRU [16, 17]:

$$z_t = \sigma(U_z \, x_t + W_z \, h_{t-1}) \quad (9)$$

$$r_t = \sigma(U_r \, x_t + W_r \, h_{t-1}) \quad (10)$$

$$c_t = tanh(U_c \, x_t + W_c(h_{t-1} \otimes r_t)) \quad (11)$$

$$h_t = (z_t \otimes c_t) \oplus ((1 - z_t) \otimes (h_{t-1}) \quad (12)$$

GRUs and LSTMs have similar performance but GRUs need less time to train as they utilize fewer parameters. Besides, they require less data for generalization. However, if there is sufficient data and computational power LSTMs can show more enhanced results [16, 25].

V. Related Work

RNNs, LSTMs, and GRUs have been applied in different NLP tasks as sentiment analysis, question answering, text generation, summarization, and machine translation. The application exploits the capability of RNNs to manipulate inputs composed of sequences of words or characters [17, 29]. RNNs can process sequences in both input and output or only one of them and so they are arranged in different topologies according to the investigated problem [16].

An RNN network was trained in a hybrid methodology that combined lexicon and deep learning approache to classify the sentiment of Arabic tweets [30]. Feature vectors were built based on the word weights. Neutral-subjective weight and positive-negative weight were computed for each word using the lexicon. Besides, an SA model employed lexicon, CNN, and Bidirectional Gated

Recurrent Unit (Bi-GRU) were proposed in [31]. The lexicon weights were used to weigh the word embedding vectors. Based on the registered results the combined CNN-GRU network has been able to extract both sentiment and context features from product reviews.

The deep architectures CNN, LSTM, Bi-LSTM, and GRU were experimented with using a word embedding and character embedding for sentiment categorization in [29]. Bi-LSTM showed the best performance using word embedding whereas CNN reported the best performance using character embedding. The results were further enhanced by combining features extracted from character CNN and Word Bi-LSTM. In addition, an LSTM network has been investigated to deduce the sentiment category of Arabic tweets in [32]. Opinions were represented as word sequences. The application of LSTM was based on their ability to recall long-term temporal dependencies by memorizing past contexts and linking them to the current state. Also, LSTM models have been applied to Arabic sentiment analysis in [15, 28, 33, 34].

The combination of CNN and LSTM has been implemented to predict the sentiment of Arabic text. The applied CNN-LSTM structures used one convolutional layer and one LSTM layer and employed either word embedding [35, 36, 37] or character representation [38]. Mazajak is a system for Arabic SA that employs a combined architecture of CNN and LSTM [37]. Word2vec embedding was trained on multiple datasets to capture different styles of dialectal Arabic. A combined CNN-LSTM model utilized character N-Gram, character, and word features for SA have been studied in [36]. Character representation generated rich features for manipulating short text as tweets. The model achieved more boosted accuracy on three benchmarking datasets. Word2Vec, FastText, and AraVec word embedding methods have been evaluated. Results highlighted that the highest performance is realized for the dataset with the largest size.

Moreover, GRUs have been investigated in [14, 39] for Arabic sentiment identification. LSTM and GRU were used to predict the sentiment category of Arabic microblogs depending on Emoji features in [14]. Results reported that LSTM and GRU classifiers performed better than other implemented classifiers. Moreover, religious hate speech has been detected implementing a GRU model and word embedding [39]. A multi-dialectal Arabic corpus of tweets was used to train word embedding. GRU reported improved performance compared to lexicon-based and machine learning classifiers.

Two deep CNNs were applied for Arabic sentiment analysis employing character-level features in [40]. A large dataset was built from free available SA datasets to train the models. The dataset maintains opinions from different domains expressed in different Arabic forms (Modern Standard, Dialectal). Multiple model structures with a different number of feature maps were tested. Results showed a prominent performance of the CNNs on the hybrid dataset.

Most implementations of LSTMs and GRUs have applied word embedding to encode words by real value vectors. Besides, CNN-LSTM applied for Arabic used only one convolutional and one LSTM layer. In the following investigation, multilayers of LSTM, GRU, CNN-LSTM, and CNN-GRU are used for feature extraction from character representation.

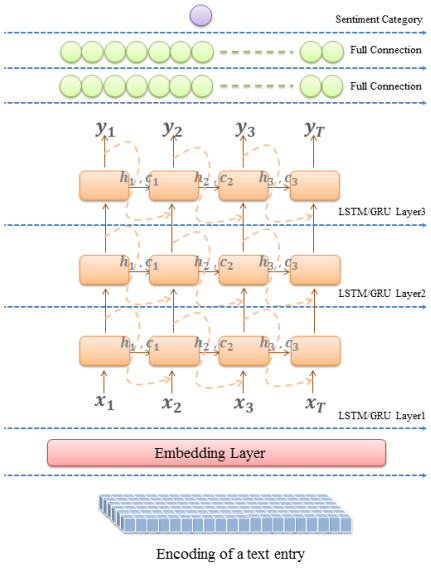


Fig. 1. LSTM/GRU NETWORK ARCHITECTURE

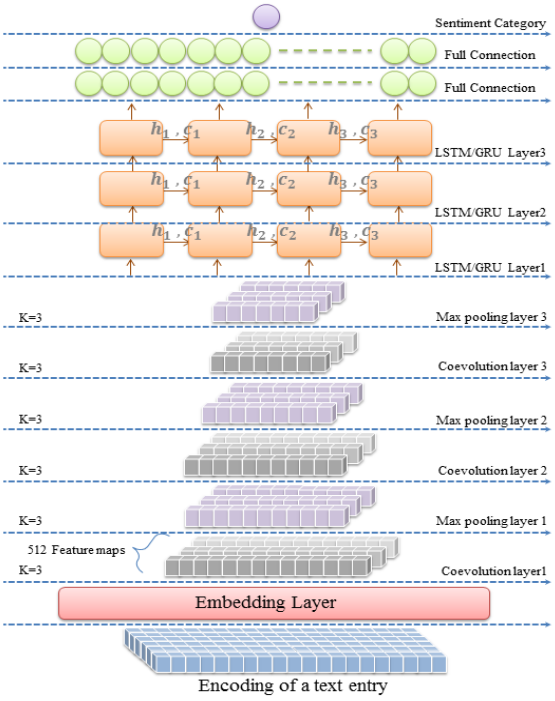


Fig. 2. CNN-LSTM/CNN-GRU NETWORK ARCHITECTURE

VI. APPLIED DEEP LEARNING MODELS

The implemented models employ deep layers for feature learning from character-level representation.

A. Network Design

LSTM and GRU models follow the structure shown in Fig. 1. The model is constructed of a character embedding layer that converts encoded text entries into a vector representation. Feature extraction layers are three LSTM or GRU layers following the embedding layer. Multiple layers enable the model to extract different levels of

feature abstraction. Each layer includes (110) cells. The final classification layers are three fully connected layers and two dropout layers follow the first and the second dense layers.

CNN-LSTM and CNN-GRU model structure is described in Fig. 2. The model is composed of three convolutional layers and three max-pooling layers arranged alternately. The convolution and pooling layers are followed by three LSTM or GRU layers. The combined hybrid layers are mounted between the embedding layer and the classification layers.

B. Network settings

Each opinion entry is represented as a sequence of characters. The character vocabulary is all the characters that appear in the dataset (Arabic characters, Arabic numbers, English characters, English numbers, Emoji, Emoticons, special characters). Text samples are quantized using a vocabulary dictionary of (746) characters. Opinion samples are encoded as sequences of length equal (1014) characters. Training is executed as in [40]. Python, Keras, and Tensorflow are used for the model application. LSTM and GRU layers are trained on CUDA9 and CUDNN7 for acceleration. The implementation is executed on NVIDIA GEFORCE GTX 1070 GPU.

VII. EXPERIMENTS AND RESULTS

A. Data preparation and preprocessing

The dataset used for training the LSTM and GRU models is the combined corpus described in [40]. The corpus merges thirteen sets from free accessible sentiment analysis corpora. The raw set contains (92492) samples. Text entries are composed in the dialectal and modern standard Arabic. Opinions in the corpora belong to various domains as tweets, product reviews, restaurant reviews, hotel reviews, book reviews, and movie reviews.

Nearly 77 percent of the dataset is positive instances and only 23 percent is negative instances. To account for the distribution bias in the training data a balancing preprocessing is applied as unbalanced sets are considered the main source of performance bias [41, 42]. A balanced dataset is generated by random oversampling of the minority class. Random oversampling modifies the data distribution by duplicating randomly selected samples. It has been applied for sentiment analysis using RNN, GRU, LSTM, and Bi-directional LSTM, [43]. It was proved that oversampling improved the model performance. To further mitigate bias in the implementation no preprocessing is applied to the dataset and the features vocabulary includes all the characters that appear in the dataset [41, 42]. TABLE I shows the reported accuracy of applying different traditional machine learning algorithms.

B. Results analysis

Accuracy, precision, recall, and F1-score are used to assess the efficiency of the proposed models. Accuracy is the percentage of all correctly predicted samples. Precision is the percent of the predicted positive entries that are positive. The recall is the percent of the actual positive entries that were predicted correctly as positive. F1-score is a measure of the model performance based on precision and recall [29]. A perfect F1 score equals one, whereas the worst is zero. Low false positives and false negatives are the two effective factors to get a high F1 value. The proposed models show high F1 values which indicate the ability to discriminate both polarity classes.

The GRU model has shown higher performance compared to the LSTM model and the deep convolutional networks applied in [40]. However, the LSTM and GRU models share the same structure and depth, the three GRU layers are more able to learn and extract character-level features. GRUs can realize a good performance on small datasets and has a higher capability of generalization. Besides, they can combat overfitting when processing small datasets [39]. As deep LSTM requires a large dataset, the LSTM model may show more enhanced measures with a larger dataset.

TABLE I. MACHINE LEARNING CLASSIFIERS ACCURACY

Classifier	TF	Unigram	Bigram
Nearest Neighbors	78.30	65.35	74.58
Support Vector Machine	85.85	77.11	76.88
Decision Tree	80.20	79.57	79.07
Random Forest	83.77	83.87	83.05
Multinomial Naive Bayes	86.15	78.29	77.75
Logistic Regression	85.58	80.21	76.90
Bernoulli Naive Bayes	81.16	81.16	76.99

TABLE II. LSTM, GRU, CNN-LSTM, AND CNN-GRU ACCURACY

Architecture	Accuracy
[40] 256 Feature Maps	92.50
[40] 1024 Feature Maps	94.12
[40] 2*(64, 128, 256, 512) Feature Maps	93.88
[40] (64, 128, 256, 512, 1024) Feature Maps	93.85
Deep LSTM	92.52
Deep GRU	94.50
Deep CNN-LSTM	95.14
Deep CNN-GRU	95.08

TABLE III. CONFUSION MATRICES OF LSTM, GRU, CNN-LSTM, AND CNN-GRU

		Precision	*Recall*	*F1-score*
LSTM	***Negative***	0.89	0.97	0.93
	Positive	0.97	0.88	0.92
	Average	0.93	0.93	0.93
		Precision	***Recall***	***F1-score***
GRU	***Negative***	0.91	0.98	0.95
	Positive	0.98	0.91	0.94
	Average	0.95	0.94	0.94
		Precision	***Recall***	***F1-score***
CNN-LSTM	***Negative***	0.92	0.98	0.95
	Positive	0.98	0.92	0.95
	Average	0.95	0.95	0.95
		Precision	***Recall***	***F1-score***
CNN-GRU	***Negative***	0.92	0.98	0.95
	Positive	0.98	0.92	0.95
	Average	0.95	0.95	0.95

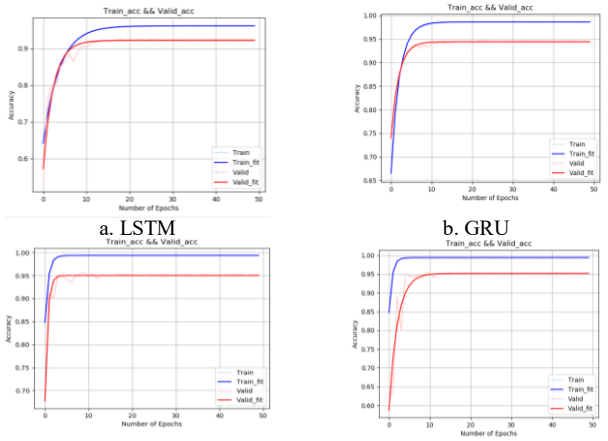
On the other hand, both hybrid structures reported higher performance compared to one type structure. CNN-

LSTM recorded accuracy equals 95.14 and CNN-GRU recorded 95.08. The model exploits the CNN feature extraction capability and LSTM/GRU power to detect long-term hidden relations between input components. The one-dimensional CNN layers keep sequential locality encountered in the ordered data elements. And hence, the subsequent LSTM/GRU layers could capture sentiment features. The registered accuracy of the proposed models and the related literature is stated in TABLE II.

Considering the positive class, recall or sensitivity measures the model's ability to cover the actual positive samples. Positive recall has reached 0.92 with the CNN-LSTM and CNN-GRU based architectures. The precision or confidence that measures the true positive accuracy has registered 0.98 with the GRU, CNN-LSTM, and CNN-GRU architectures. The same statistics are computed for the negative class by predicting the opposite case. The negative class recall measures the model coverage of the actual negative samples that reached 0.98 with the GRU, CNN-LSTM, and CNN-GRU based structures. The precision that measures the true negative accuracy has reported 0.92 with the CNN-LSTM, and CNN-GRU structures. Precision, recall, and F1-score of the models are reported in TABLE III.

The LSTM model demonstrated slower learning compared to the GRU, CNN-LSTM, and CNN-GRU models. In addition, CNN-LSTM, and CNN-GRU reached a higher accuracy within several epochs less than both LSTM and GRU. Training and validation accuracies of the applied models are shown in Fig. 3. The confusion matrix graphs illustrated in Fig. 4 show high values of the identified true positive and true negative samples. The results emphasize the efficiency of the applied bias handling technique. The class-based performance measured by class precision, recall, and F1-score are closer for CNN-LSTM and CNN-GRU as stated in TABLE III.

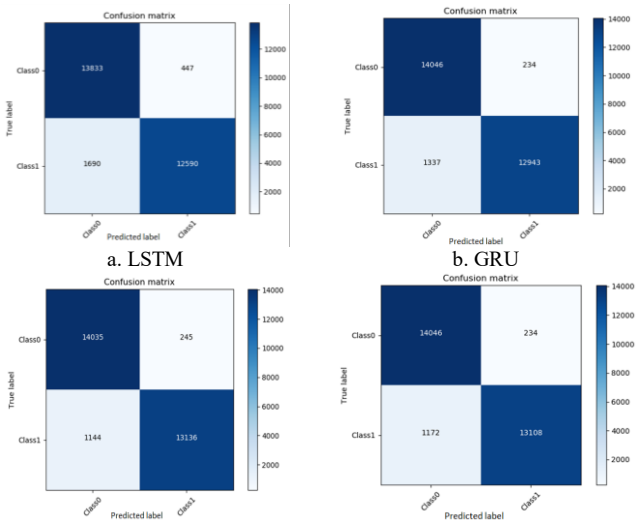
A data sample from the training set expressed in Arabic and its translation to English is indicted in TABLE IV. Fig. 5 displays the extracted features from the sample by the deep LSTM, GRU, CNN-LSTM, and CNN-GRU. The LSTM model highlighted more features compared to the GRU. Along with the theoretical basis of LSTM, it is a more powerful structure but requires more data to train. Whereas, the GRU light structure could extract more discriminative features. Besides, the CNN-LSTM and CNN-GRU captured more condensed features.



a. LSTM b. GRU

c. CNN-LSTM d. CNN-GRU

Fig. 3. TRAINING ACCURACY



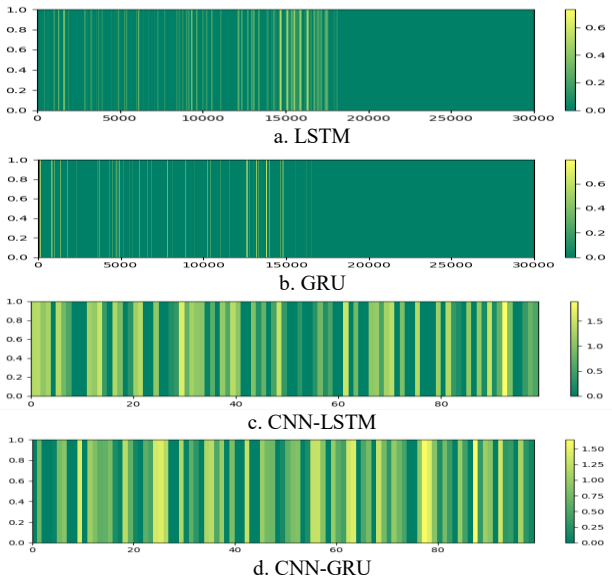
a. LSTM b. GRU

c. CNN-LSTM d. CNN-GRU

Fig. 4. CONFUSION MATRIX

TABLE IV. EXAMPLE OF OPINION TEXT

Arabic opinion sample	"إستخدام مبدع للألفاظ. و تطرق لمناطق و تساؤلات تدور فى كل النفوس و العقول. ثم إجابات بسيطة و مقنعة لكل هذه التساؤلات. فعلا إستمتعت بقراءة هذا الكتاب"
English translation	"Creative use of words. It touched upon areas and questions that revolve in all souls and minds. Simple and convincing answers to all these questions. I really enjoyed reading this book."



a. LSTM

b. GRU

c. CNN-LSTM

d. CNN-GRU

Fig. 5. EXTRACTED FEATURES

VIII. CONCLUSION

Deep architectures have proved to be efficient feature extractors, however; they rely on intensive computations and large datasets. Recently, deep learning LSTM, GRU, Bi-LSTM, and CNN have been extensively investigated in sentiment polarity detection. The applied deep LSTM and GRU models have extracted relevant features from the user-generated text in the raw state. Deep LSTM and GRU layers enabled the model to learn multiple levels of abstraction from the input over time. Furthermore, discriminating features have been detected from the character level representation of the input text sequence. The GRU model reported enhanced performance compared to the LSTM with the same structure. Also, the GRU model outperformed the referred CNN deep networks. In addition, deep hybrid networks realized the highest performance measures compared to the applied models. Combining CNN and LSTM/GRU showed more boosted performance which supports the application for other NLP tasks. Also, the applied models identified both classes with comparable performance measures. For future work, hybrid architectures that combine different deep network structures can be implemented and assessed. The performance of hybrid architectures may be studied using various word representation approaches and further analyzed based on character level representation in other NLP tasks.

REFERENCES

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," Journal of Machine Learning Research, pp. 2493-2537, 2011.

[2] Y. Chen, "Convolutional neural network for sentence classification," Master thesis, Dept. of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2015.

[3] S. Dhuria, "Natural language processing: An approach to parsing and semantic analysis," International Journal of New Innovations in Engineering and Technology, vol. 3, no. 1, pp. 51-55, 2015.

[4] A.R. Pal and D. Saha, "Word sense disambiguation: A survey," International Journal of Control Theory and Computer Modeling (IJCTCM), vol. 5, no. 3, pp. 1-16, 2015.

[5] I. Sharma and P.K. Singh, "A Survey on Anaphora Resolution," IJCA Proceedings on Recent Innovations in Computer Science and Information Technology (RICSIT 2016), no. 1, pp. 5-7, 2016.

[6] V.S. Jagtap and K. Pawar, "Analysis of different approaches to sentence-level sentiment classification," International Journal of Scientific Engineering and Technology, vol. 2, no. 3, pp. 164-170, 2013.

[7] M.A. Ibrahim and N. Salim, "Sentiment Analysis of Arabic Tweets: With Special Reference Restaurant Tweets," IJCST, vol. 4, no. 3, pp. 173-179, 2016.

[8] A.F. El Gohary, T.I. Sultan, M.A. Hana, and M.M. El Dosoky, "A computational approach for analyzing and detecting emotions in Arabic text," International Journal of Engineering Research and Applications (IJERA), vol. 3, no. 3, pp. 100-107, 2013.

[9] S. Al-Saaqa, H. Abdel-Nabi, and A. Awajan, "A Survey of Textual Emotion Detection," In the 8th International Conference on Computer Science and Information Technology (CSIT), IEEE, July 11, Amman, Jordan, pp. 136-142, 2018.

[10] N. Gupta, "Learning Distributed Document Representations for Multi-label Document Categorization," Master thesis, INDIAN institute of technology, Kanpur, India, Dept. of Electrical Engineering ,2015.

[11] M. El-Haj, U. Kruschwitz, and C. Fox, "Using Mechanical Turk to create a corpus of Arabic summaries," In Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages workshop. the 7th International Language Resources and Evaluation Conference (LREC 2010), May 19, Valletta, Malta, pp. 36-39, 2010.

[12] A. Dahou, M.A. Elaziz, J. Zhou, and S. Xiong, "Arabic sentiment classification using convolutional neural network and differential evolution algorithm," Computational intelligence and neuroscience, vol. 2019, no. 2537689, pp. 1-16, 2019.

[13] S. Dargan, M. Kumar, M.R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," Archives of Computational Methods in Engineering, vol. 2020, no. 27, pp. 1071–1092, 2020.

[14] S. Al-Azani and E.-S. El-Alfy, "Emojis-based sentiment classification of Arabic microblogs using deep recurrent neural networks," In Proceedings of the 2018 International Conference on Computing Sciences and Engineering (ICCSE), pp. 1–6, IEEE, Kuwait, 2018.

[15] M. Abbes, Z. Kechaou, and A. M. Alimi, "Enhanced deep learning models for sentiment analysis in Arab social media," In Proceedings of the International Conference on Neural Information Processing, pp. 667–676, Springer, China, 2017.

[16] A. Gulli and S. Pal, "Deep learning with Keras," Packt Publishing Ltd, 2017.

[17] O. Calin, "Deep Learning Architectures," Springer International Publishing, 2020.

[18] A. Yadav and D.K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," Artificial Intelligence Review, vol. 53, no. 6, pp. 4335-4385, 2020.

[19] B. Jang, M. Kim, G. Harerimana, S.U. Kang, and J.W Kim, "Bi-LSTM model to increase accuracy in text classification: combining Word2vec CNN and attention mechanism," Applied Sciences, vol. 10, no. 17, p. 5841, 2020.

[20] M.V. Mäntylä, D. Graziotin, and M. Kuutila, "The evolution of sentiment analysis—A review of research topics, venues, and top-cited papers," Computer Science Review, vol. 27, pp. 16-32, 2018.

[21] P. Borele and D.A. Borikar, "A Survey on Evaluating Sentiments by Using Artificial Neural Network," International Research Journal of Engineering and Technology (IRJET).vol. 3, no. 2, pp. 1402-1406, 2016.

[22] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. 1253, 2018.

[23] B.S. Harish, D.S. Guru, and S. Manjunath, "Representation and classification of text documents: A brief review," IJCA, Special Issue on RTIPPR, vol. 2, pp. 110-119, 2010.

[24] K. Grzegorczyk, "Vector representations of text data in deep learning," Doctoral thesis, AGH University of Science and Technology, Faculty of Computer Science, 2018.

[25] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidiki, M.S. Nasrin, M. Hasan, B.C. Van Essen, A.A. Awwal, and V.K. Asari, "A state-of-the-art survey on deep learning theory and architectures," Electronics, vol. 8. no. 3, p.292, 2019.

[26] S. Krig, "Feature learning and deep learning architecture survey," In Computer Vision Metrics, pp. 375-514, Springer, Cham, 2016.

[27] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification", Advances in neural information processing systems, December 7, Montreal, Quebec, Canada, pp. 649-657, 2015.

[28] A. Mohammed and R. Kora, "Deep Learning Approaches For Arabic Sentiment Analysis," Springer journal: Social Network Analysis and Mining, vol. 9, no. 52, pp. 1869-5469, 2019.

[29] M.U. Salur and I. Aydin, "A Novel Hybrid Deep Learning Model for Sentiment Classification," IEEE Access, vol. 8, pp. 58080-58093, 2020.

[30] K. Elshakankery and M. F. Ahmed, "HILATSA: A hybrid Incremental learning approach for Arabic tweets sentiment analysis," Egypt Informatics J., vol. 20, no. 3, pp. 163–171, 2019.

[31] L. Yang, Y. Li, J. Wang, and R.S. Sherratt, "Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning," IEEE Access, vol. 8, pp. 23522-23530, 2020.

[32] M. Heikal, M. Torki, and N. El-Makky, "Sentiment Analysis Of Arabic Tweets Using Deep Learning," Procedia Computer Science, vol. 142, pp. 114-122, 2018.

[33] A. Oussous, F.Z. Benjelloun, A.A. Lahcen, and S. Belfkih, "ASA: A framework for Arabic sentiment analysis," Journal of Information Science, vol. 46, no. 4, pp. 544-559, 2020.

[34] A.Q. Albayati, A.S. Al-Araji, and S.H. Ameen, "Arabic Sentiment Analysis (ASA) Using Deep Learning Approach," Journal of Engineering, vol. 26, no. 6, pp. 85-93, 2020.

[35] S. Al-Azani and E.-S. M. El-Alfy, "Hybrid deep learning for sentiment polarity determination of Arabic microblogs," In International Conference on Neural Information Processing, November 14, Guangzhou, China, pp. 491–500, 2017.

[36] A.H. Ombabi, W. Ouarda, and A.M. Alimi, "Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual

information shared in social networks," Social Network Analysis and Mining, vol.10, no. 1, pp.1-13, 2020.
[37] I.A. Farha and W. Magdy, "Mazajak: An Online Arabic Sentiment Analyser," In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Italy, pp. 192-198, 2019.
[38] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," International cross-domain conference for machine learning and knowledge extraction, August 27, Hamburg, Germany, pp. 179–191, 2018.
[39] N. Albadi, M. Kurdi, and S. Mishra, "Investigating the effect of combining GRU neural networks with handcrafted features for religious hatred detection on Arabic Twitter space," Social Network Analysis and Mining, vol. 9, no. 1, p.41, 2019.
[40] E. Omara, M. Mousa, and N. Ismail, "Deep Convolutional Network For Arabic Sentiment Analysis," International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC), IEEE, pp. 155-159, 2018.
[41] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, and I. Kompatsiaris, "Bias in data‐driven artificial intelligence systems-An introductory survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 10, no. 3, p.1356, 2020.
[42] D. Roselli, J. Matthews, and N. Talagala, "Managing bias in AI," In Companion Proceedings of The 2019 World Wide Web Conference, pp. 539-544, 2019.
[43] A. Mukherjee, S. Mukhopadhyay, P.K. Panigrahi, and S. Goswami, "Utilization of oversampling for multiclass sentiment analysis on amazon review dataset," In 2019 IEEE 10th International Conference on Awareness Science and Technology (ICAST), IEEE, pp. 1-6, 2019.