Neural Computing and Applications (2022) 34:1135–1159
https://doi.org/10.1007/s00521-021-06390-z

ORIGINAL ARTICLE

Arabic text classification: the need for multi-labeling systems

Hozayfa El Rifai[1] · Leen Al Qadi[1] · Ashraf Elnagar[1]

Received: 27 August 2020 / Accepted: 26 July 2021 / Published online: 1 September 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
The process of tagging a given text or document with suitable labels is known as text categorization or classification. The aim of this work is to automatically tag a news article based on its vocabulary features. To accomplish this objective, 2 large datasets have been constructed from various Arabic news portals. The first dataset contains of 90k single-labeled articles from 4 domains (Business, Middle East, Technology and Sports). The second dataset has over 290 k multi-tagged articles. To examine the single-label dataset, we employed an array of ten shallow learning classifiers. Furthermore, we added an ensemble model that adopts the majority-voting technique of all studied classifiers. The performance of the classifiers on the first dataset ranged between 87.7% (AdaBoost) and 97.9% (SVM). Analyzing some of the misclassified articles confirmed the need for a multi-label opposed to single-label categorization for better classification results. For the second dataset, we tested both shallow learning and deep learning multi-labeling approaches. A custom accuracy metric, designed for the multi-labeling task, has been developed for performance evaluation along with hamming loss metric. Firstly, we used classifiers that were compatible with multi-labeling tasks such as Logistic Regression and XGBoost, by wrapping each in a OneVsRest classifier. XGBoost gave the higher accuracy, scoring 84.7%, while Logistic Regression scored 81.3%. Secondly, ten neural networks were constructed (CNN, CLSTM, LSTM, BILSTM, GRU, CGRU, BIGRU, HANGRU, CRF-BILSTM and HANLSTM). CGRU proved to be the best multi-labeling classifier scoring an accuracy of 94.85%, higher than the rest of the classifies.

Keywords Arabic text classification · Single-label classification · Multi-label classification · Arabic datasets · Shallow learning classifiers · Deep learning classifiers

1 Introduction

Large numbers of repositories live online as a result of the heavy usage of Web 2.0 and the Internet, which leads to a need and demand for automatic classification methods. Almost 80% of data is textual and unstructured but considered an extremely valuable and rich source of information. Machine learning algorithms have been proved helpful in cutting the time needed to extract insights and organize massive chunks of data.

One of the fundamental tasks in natural language processing (NLP) is text classification. It is used to assign labels to textual data based on its context. Automating the process simplifies classifying documents, helps in standardizing the platform, and makes searching for specific information straightforward and possible.

Manually classifying documents by experts is not as efficient as it used to be due to their increasing amount. This is where machine learning algorithms come into play, as an alternative to conventional ways. They produce faster and more fruitful results. Several examples and applications of text classification have been explored such as language identification [38], dialect identification [19], sentiment analysis [16, 17, 22–24] and spam filtering [2, 37].

Structuring data using machine learning are becoming essential in the business field. It helps in detecting new

✉ Ashraf Elnagar
ashraf@sharjah.ac.ae

Hozayfa El Rifai
u16103377@sharjah.ac.ae

Leen Al Qadi
u16103630@sharjah.ac.ae

[1] Department of Computer Science, University of Sharjah, Sharjah, UAE

patterns and trends and identifies relationships between seemingly unrelated data. For example, marketers can gather and review keywords used by other firms in the field. For Arabic NLP, this is still a challenging task among others [29]. It is a very complex, morphological-derived language, and it is the mother tongue of over 300M people. The research done in the field of Arabic computational linguistics is impressively increasing in the last decade, but the work still has room to expand and branch out.

Arabic has been reported by the Internet World Stats to be the 4th most popular language online, with over 225k users, representing 5.2% of all Internet users as of April 2019. They also show that it has the highest growth rate in the number of users in the last 19 years, achieving 8,917.3%.

Working with Arabic text is different from working with English text. It is more challenging for several reasons which include the following: (1) Forms: Arabic has three different forms (Classical, Modern Standard Arabic (MSA), and Dialectal), (2) Vocabulary size: Arabic language has around 12.3 million words compared to 600,000 words in English, (3) Alphabet set: the character set has 28 consonants and 8 vowels. Besides, when writing cursive script several characters have different shape-forms. The Arabic text is written from right to left, (4) Grammar: Arabic assigns words, verbs, and pronouns to a gender. It also has singular and plural forms for both male and female. Further, conjugated verbs in Arabic are different, (5) Vowels: a significant difference is in the quality and length of the vowels. Arabic generally uses diphthongs and long vowels as in-fixes, and (6) Sentence structure: Arabic has verbal and nominal sentences. A nominal sentence does not require a verb.

In this work, we first introduce a newly built single-labeled dataset of Arabic news articles, collected from several portals to aid our research. Several classifier models are trained to predict the single class a proposed article belongs to. Moreover, a voting classifier has been implemented, considering the best predicting classifiers, with the highest accuracy percentage.

An Arabic news article single-labeling classification excerpts from text the linguistic features using the TF-IDF technique. In the training phase, each article is turned into a feature vector, and then, the system identifies the most common features under each category. This way, when the classifier encounters a new article, it will attempt at predicting the relevant category based on its features vectors.

We propose a multi-class classifier, to label an Arabic news articles under an appropriate class out of 4 classes. We used a supervised approach for text classification. We tested different vectorization methods to seek the best, and to see the effect they have on the accuracy percentages. In addition, we investigated the possibility of using a customized list of stop words as a replacement for using the NLTK list.

Looking at the misclassified articles by the classifying system, we decide to construct a new Arabic multi-labeled dataset for the purpose of assigning the articles to multi-labels instead. Two approaches were tested. Firstly, we implemented two classical classifiers made compatible with the task of multi-labeling. Second, we built ten neural networks with unique architectures to test the effectiveness of deep learning techniques. For the first approach, we used the TF-IDF technique for feature extract. Both classifiers need to be wrapped in a OneVsRest Classifier, to convert the classification problem to sub-problems. For the second approach, we used the tokenizer provided by Keras before training.

We offer a multi-label multi-class text categorization system that is capable at assigning an article with multiple labels out of 21 labels. We evaluate and compare the two approaches using the custom accuracy metric specific to the multi-labeling system, along with hamming-loss scores. This work is an extension of our work [4, 5] on single-label classification.

To summarize, in this work, we propose two new large datasets for Arabic news articles tagging. One dataset is dedicated for single-label classification, while the other dataset is dedicated for multi-label classification. Both datasets are new to the Arabic computational linguistics field and shall serve the need for such rich datasets. Furthermore, we demonstrate the validity and efficacy of these proposed datasets by studying the performance of several shallow as well as deep learning models to classify Arabic text. This is a comprehensive study on the task of Arabic news classification, which is needed to fill this research gap. In conclusion, the contributions of the work are:

- Two large Arabic datasets, which tat are comprised of news articles spanning several topics, are properly annotated for Arabic text classification. The datasets shall be made available for researchers in the Arabic natural language processing field.
- A rigorous investigation of several shallow and deep learning classification models is carried out for the Arabic text classification task to choose the best models.
- Considerable experiments are conducted to confirm the fitness of the proposed datasets and the classification models.
- Fine-tuning of the models as well as the utilization of word embedding is performed to achieve solid performance.

The paper is organized as follows: The literature review is presented in Section 2. Section 3 demonstrates the dataset.

Sections 4 and 5 detail the proposed classification process. The results and discussion are presented in Sect. 6. At last, we present our conclusions in Sect. 7.

2 Literature review

Many surveys and papers shed light on the different approaches used for English text categorization and discuss existing literature [1, 18, 36]. For Arabic text classification, surveys like [7, 33] also exist. Some researchers investigated the classification task on other languages such as Portuguese. In [28], they used an SVM classifier on an English dataset and on a Portuguese dataset. They found that the Portuguese language required paying more attention to the document representation like semantic/syntactic information and word order.

Arabic text classification research and the goal to enrich the Arabic corpus are slowly becoming a priority in the research community. In [31], the authors believe that many of the available datasets are not appropriate for classification, either because the classes are not defined well, or there are not any defined classes like in the 1.5 billion words Arabic Corpus [11]. The authors also introduce ’NADA,’ a new filtered and preprocessed corpus, that combine already existing corpora DAA and OSAC. ’NADA’ contains 13,066 documents belonging to 10 categories in total. With regard to the number of labels, we believe that the corpus is too small.

Recent research papers focusing more on Arabic text classification (ATC) are emerging. The author in [20] used a dataset of articles collected from (aljazeera.com), to compare the performance of 6 different classifiers. Under the same environmental settings, Naive Bayes was the best classifier, regardless of feature selection methods.

Many papers experiment with feature selection methods. In [8], they studied the effect of using uni-grams and bi-grams, experimenting with the KNN classifier. In [39], they reported that using an SVM classifier, for ATC, outperforms other classifiers. An experiment on 4 classifiers while by means of 2 feature selection techniques (information gain and chi-squared) was conducted on a BBC Arabic dataset in [41]. Lastly, in [31], the authors present a new feature selection method for ATC, and it outperforms five other approaches, testing them using the SVM classifier.

Many authors reported results of supervised classical machine learning algorithms such as NB [12, 14, 21, 40], SVM [3, 12, 27, 32], Decision Tree [3, 30, 42], KNN [14, 32].

Several others preferred to work with deep learning techniques and experimented with neural networks like in [9, 10, 15, 25] to tackle the single-label classification problem. Authors pursuing better results using a different approach such as [15] have used a convolution neural network for Arabic text classification and achieved better results than Logistic Regression and SVM. In [10], the authors proved that using feature reduction techniques with an ANN model achieves higher results than a basic ANN model. An extended work to address the multi-labeling classification task using a variety of deep learning models is studied in [26]. In this work, we show a more comprehensive study by including classical machine learning algorithms, which produce superb results as well.

All the above references worked on the single-labeling task of Arabic text. Nonetheless, the need for integrating multi-labeling is becoming essential. A vast set of news article span more than one major topic. For example, a news article that talks about covid-19 (medical domain) and its impact on the economy should be tagged with both labels rather than only one. Multi-labeling would resolve the intersection of multiple domains instead of just selecting one. In fact, more electronic news portals are tagging each news article with multiple tags (keywords). This process is usually carried out by humans. Therefore, the need for an automated tagging system is becoming a necessity. While multi-labeling task is well researched for the English language (for example, see [13, 44]), it is under-researched for Arabic language. This work helps to bridge this gap in the Arabic computational linguistic field. In the sequel, we describe few studies on this task for Arabic.

Shehab et al. [43] investigated the multi-label classification task using three machine learning classifiers, which are Decision Trees (DT), Rain Forest (RF), and KNN. The results show that DT outperforms the other 2 classifiers. This is a limited study; there are more robust classifiers that can outperform DT as we show in our work. Hmeidi et al. [34] used a lexicon-based system to classify Arabic documents. The dataset has 8,800 multi-label documents collected from BBC Arabic. Several single-label and multi-label lexicons were produced to tackle the problem. The dataset is relatively small to handle effectively multi-label tagging. Besides, scalability is a major concern with lexicon-based methods. Both of these works used hamming loss metric as an evaluating metric, along with precision and recall.

Al-Salemi et al. [6] proposed a new dataset gathered from RT-news (RTANews) website for multi-labeling of Arabic news articles. They explored 4 transformation-based algorithms: Binary Relevance, Classifier Chains, Calibrated Ranking by Pairwise Comparison and Label Powerset. They used 3 classifiers, namely SVM, KNN, and RF. They reported that RF and SVM produced the best results. However, the dataset has 87% of the documents was tagged with a single label. As a result, the dataset is biased toward single-label rather than multi-label classification. This shortcoming would heavily impact the performance of the proposed algorithms on another balanced dataset.

It is clear that the accuracy and the general performance are highly dependable on the quality of the collected data and by the feature representation method. The more redundant features we have, the less accurate the classification is. Therefore, we introduce new rich and representative datasets for treating the problem of both single-label and multi-label Arabic documents classification. We truly believe that the datasets would serve as benchmarks. In contrast with the existing research works on this task, we provide a thorough examination of several shallow and deep learning algorithms to robustly solve the automatic tagging of Arabic news articles.

3 Datasets

We visited 7 popular news portals (arabic.rt.com, youm7.com, cnbcarabia.com, beinsports.com, arabic.cnn.com, skynewsarabic.com and tech-wd.com), to collect the articles from. We used (Python Scrapy) library to scrape the articles. The single labeled dataset contains 89,189 Arabic documents (approximately 32.5 million words). The articles of the dataset are categorized under four main classes: ['Sports,' 'Middle East politics,' 'Business' and 'Technology'].

All the collected articles are written in Modern Standard Arabic (MSA), with no dialects. The articles are grouped in one corpus.

Table 1 and Fig. 1 describe the distribution of the articles under the 4 categories in the dataset. In average, we scraped around 22k articles for each category. We made sure to avoid bias by constructing a balanced dataset.

Table 1 Number or documents collected from 7 news portals

Websites	Classes	Articles count
Sky News Arabia	Sports	7923
CNN Arabia	Sports	3800
	Tech	1680
	Middle East	21,516
	Business	3908
Bein Sports	Sports	6603
Tech-wd	Tech	23,682
Arabic RT	Business	896
Youm7	Business	14,478
CNBC Arabia	Business	4653

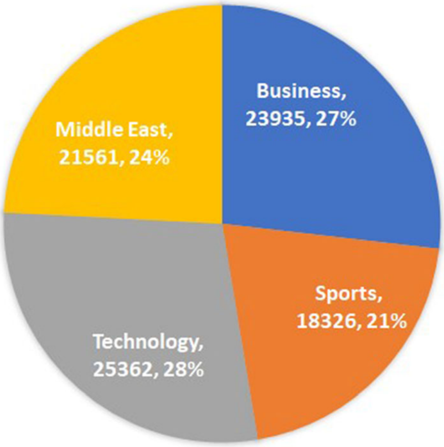


Fig. 1 Some statistics on the single-label dataset categories

As for the multi-label dataset, we collected this dataset using (Python Scrapy, Selenium and BeautifulSoup) from ten different websites listed in Table 2. The articles in this dataset, all belong to one of the 4 classes: [Middle East, Business, Technology, Sports], in additional to hundreds of tags. It consists of 293,363 multi-tagged articles written in (MSA). Figure2 shows the distribution of the articles in the main categories.

4 Single-label classification

4.1 Text features

Machine learning algorithms cannot process text directly, and to solve this, we represent the text in numerical vectors. Words of the articles represent categorical features, and each sentence will be presented by one vector. This process is called vectorization. Two of the most common techniques used in text vectorization are countVectorizer and TF-IDFVectorizer. Both of these vectorizers are used to represent textual data in vector format. While countVectorizer keeps track of the number of tokens (i.e., features) encountered in a document, TF-IDFVectorizer

Table 2 Scraped news portal for the multi-label dataset

Websites scraped	
CNBC Arabia	Bein Sports
CNN Arabia	Tech-wd
Masrawy	aitnews
Youm7	Arabic RT
Al Arabiya	SkyNewsArabia

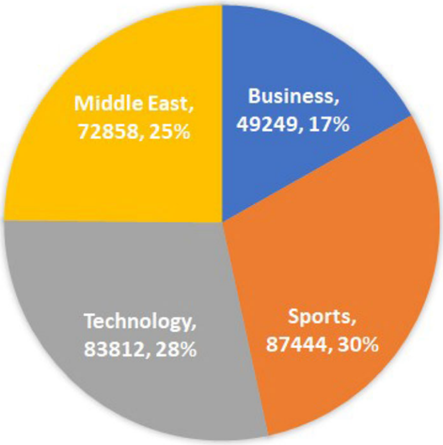


Fig. 2 Some statistics on the multi-label dataset categories

stores the weighted frequency of each token with respect to the document. TF-IDFVectorizer is favored over countVectorizer as the latter one is biased to most frequent tokens opposed to low frequent features that may be key feature in determining the document genre. To overcome this problem, we adopted TF-IDFVectorizer, which will compute the relative frequency of each feature in each document. This vectorizer computes the most common features which could identify the document main topics. However, as it is based on the bag-of-words (BoW) concept, it does not capture the semantics when compared to other models such as word embeddings. We conducted an experimental comparison between the two vectorizer methods. We used a subset of our single-labeled dataset, containing 40k articles, classified under three categories. The comparison involves using each vectorizer as the features selection technique, which shall be fed to the same classifier (SVM) to determine the document genre (single-label) for all documents in the dataset. Table 3 confirms that higher accuracy scores were produced when using the TF-IDF vectorizer as opposed to the countVectorizer.

Table 3 TF-IDF vectorizer versus count vectorizer: performance evaluation

Algorithms	TF-IDF vectorizer	Count vectorizer
LR	96.4	**97.3**
SVM	**97.5**	97.0
DT	**92.4**	91.7
MNB	91.1	**96.8**
XG	**91.2**	**91.2**
KN	**95.0**	69.9
RF	**95.1**	94.5

Tokens that appear very frequently have a less of an impact when being represented by the term frequency-inverse document frequency. This vectorizer is made up of two components:

- Term Frequency (TF): computes how many times a word appears in a given document, then adjusts the frequency taking into consideration the length of the document.
- Inverse Document Frequency (IDF): computes how common or rare a word is in the entire article set. If a word appears many times and is common, the score approaches 0, otherwise, it approaches 1.

After that, we conducted another comparison using a custom-made list of stop-words instead of the built-in list. In fact, we adopted the customized list as it reported better results. The general flow of operations of the proposed system is described in Fig. 3.

4.2 Selected classifiers

Several different supervised classifiers are used for text classification, where the main purpose is to tag an input text with the best representative label. We studied and observed the performance of ten shallow learning models, in addition to the ensemble classifier. Next, we describe all implemented algorithms:

- Logistic Regression (LR): this is a predictive model. It is a statistical learning technique used for the task of classification. Even though the name of the classifier has the word 'Regression' in it, it is used to produce discrete binary outputs.
- Multinomial Naïve Bayes (MNB): this classifier estimates the probability of each class-label, based on Bayes theorem, for some text. The result is the class-label with the highest probability score. MNB assumes the features are independent, and as a result, all features contribute equally to the computation of the predicted label.
- Decision Tree (DT): DT is basically a tree, where nodes represent features and leaves are the output labels. Branches indicate decisions and whenever a decision is answered, a new decision will be inserted recursively until a conclusion is made. Recursion is used to partition the tree into several decisions with possible results.
- Support Vector Machines (SVM): SVM is a very prevalent supervised classifier. It is non-probabilistic. SVM uses hyperplanes to segregate labels. SVM supports linear and nonlinear models. Basically, each hyperplane is expressed by the input documents (vector)

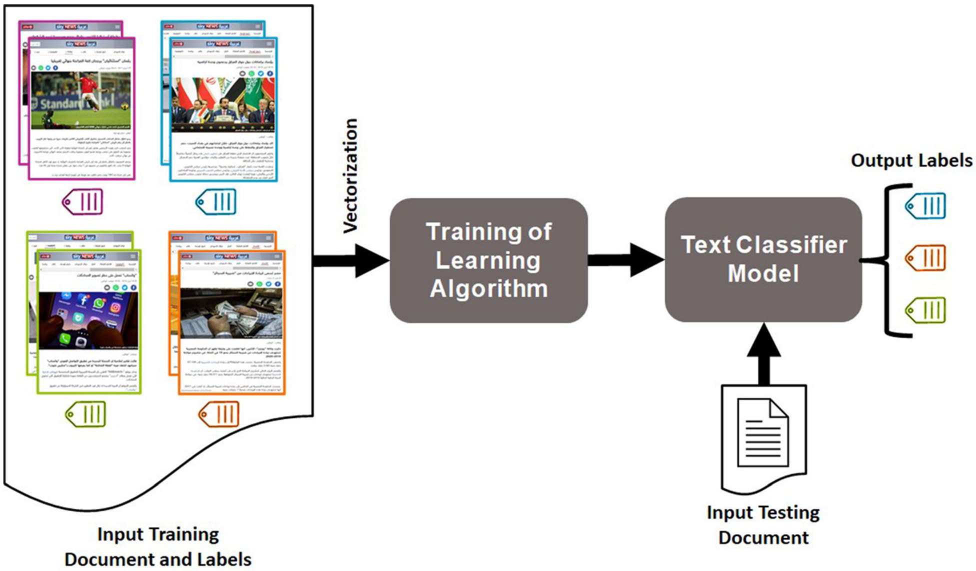


Fig. 3 Generic system flow-diagram

x satisfying $w\dot{x} - b = 0$, where w is the normal vector to the hyperplane, and b is the bias.

- Random Forest (RF): RF is a supervised learning-based classifier. This ensemble model utilizes a set of decision trees, which computes the resulting label aggregately. Strictly, the input is the documents $x_1, x_2, \ldots, x_n$ with their matching labels $y_1, y_2, \ldots, y_n$. Each decision tree f_b is trained using a random sample (X_b, Y_b), where b ranges from 1 to the total number of trees. The forecasted label shall be computed by a majority vote of all used trees.
- XGBoost Classifiers (XGB): XGB is another supervised classifier. This robust classifier became popular as a result of winning several Kaggle contests. Similar to RF, it is an ensemble classifier made of decision trees and a variant of the gradient boosting algorithm.
- Multi-layer Perceptron (MLP) Classifier: MLP is comprised of at least 3 layers of neuron nodes (input layer, hidden layers, and output layer). As for general neural networks, each node is connected to nodes in the subsequent layer. MLP utilizes a non-linear activation function to produce the resulting label.
- KNeighbors Classifier (KNN): KNN classifier determines the neighbors of an input document. The predicted class label is collectively computed by all determined neighbors, where each one votes for the closest label. The class-label with the maximum ballots is adopted. This is another major vote classifier.
- Nearest Centroid Classifier (NC): for NC, the class-label is the centroid of its data points. Given an input document then the class-label is computed based on the training examples whose mean (centroid) is closest to the input document.
- AdaBoost Classifier (ADB): ADB is a meta-estimator that starts by fitting a classifier on the training set of documents. Next, AB fits additional copies of the classifier on the training dataset but after adjusting the weights of misclassified documents such that succeeding classifiers attend to problematic cases[1].
- Ensemble/Voting Classifier (VC): VC is basically an ensemble solution. VC is packaging all preceding classifiers. Majority voting is utilized for predicting the final class label.

5 Multi-label classification systems

5.1 Classical classifiers

We selected 2 classical classifiers: [OneVsRestLogisticRegression, and OneVsRestXGBoost]. The OneVsRest Classifier decomposes the multi-label problem into multiple independent binary classification problems (one per label). Both LogisticRegression classifier and XGBoost classifier were each wrapped inside a OneVsRestClassifier. We used the TF-IDF technique to vectorize the articles, and we chose to keep the default hyperparameters for each classifier. To encode the labels, we used `MultiLabelBinarizer()` that returns the string labels assigned to each article in a one-hot encoded format.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

5.2 Deep learning classifiers

Artificial neural networks are computational models that are designed to behave similarly to the functionality of a human brain from analyzing information to deciding actions. They are implemented to solve problems that are impossible or hard to solve by human or mathematical standards. Traditionally, a neural network consists of plenty of nodes that work in parallel with each other, consisting of a layer. These layers are then connected to make up the network. Deep neural networks are neural networks with different numbers of layers between both layers, the input layer and the output layer. These additional layers are called hidden layers. Their main objective is to add more computation to the network to solve complicated tasks. These three main layers are connected to form a DNN model. These networks can be tweaked to perform well on images, audio and textual data.

In our work, we use each of convolution neural networks and recurrent neural networks models for the multi-labeled text classification. Overall, we designed ten different models. These models are described below.

Convolution neural networks (CNN) are unique classes of neural network models that are designed to identify hidden patterns and relationships in large data sets. The role of CNNs is to learn different features utilizing many filters. These features are then taken down further to learn high-order features. CNN uses a pooling layer which is main function is to reduce the spatial dimensions by performing downsampling. Applying a max-pooling layer with filters of size $2x2$ is the most common approach in pooling. Here, we perform a pooling operation that returns the maximum value on a window of size $2x2$. Another type called average-pooling works similar to the mentioned type above however, this will return only the average value instead.

Recurrent neural networks (RNN) RNN models are more powerful in use cases in which context and time are critical. Backpropagation enables looping information back through the network which empowers RNN models with processing sequential data. This helps finding correlations between dependent variables. In our work, we combine CNN and RNN to generate a new CRNN architecture. The new architecture applies a spatial dropout for the embedding layer, we then have a Conv1D layer followed by an RNN layer, we then add a global max pooling layer followed by a dropout layer to apply regularization. Finally, we have a dense layer followed by the output layer. Next, we describe the most common models of RNN.

Long short-term memory (LSTM) can be constructed by using a number of LSTM units to make up the network. They are applicable for different tasks that require sequential data such as text classification and time series forecasting. This is different from RNN since they can remember longer sequences for long periods of time. LSTM has three types of gates: the input, the output gates and the update gates. These gates determine what data to keep and what data to throw away.

Gated Recurrent Unit (GRU), although both units tackle the vanishing gradient problem, GRUs are distinguished from LSTMs by possessing a hidden state (memory). GRU utilizes 2 gates: an update gate and reset gate. The reset gate determines the amount of past information to forget, while the update gate decides what information to keep and what to not.

Hierarchical Attention Network (HAN) is extremely good at predicting the class of a given document, because it recognizes its structure. A document is constructed by first encoding the words, then encoding the sentences to form a document representation. An attention mechanism is also used in a HAN, to make sure that the context of a word or a sentence is accounted for. Words and sentences also differ in importance in regard to the main message of the document.

5.3 Proposed deep learning models

In our work, we developed ten deep learning models. We experimented the combination of both CNN and RNN in addition to CRF-CNN have been used to develop many models on different tasks. These new approaches have shown better results compared to the traditional machine learning models.

Each of these models is composed of an input embedding layer, multi hidden layers, and a dense output layer. The utilization of embedding layer helps in capture the semantic relationships between words and so it assigns similar words similar representations.

An embedding layer is internally used to convert a sparse one-hot encoded matrix to a dense vector space. This reduces the computational complexity and so the training time. A word2vec word embedding was first trained on our dataset and used in our experiment; however, it was producing poor results. Thus, we ditched it and kept using the Keras tokenizer. The size of the input is set to 200 words, and this was obtained by trying different sizes. We then design different types of deep neural network and use the best design in our work.

For the last dense layer, which is the output layer, we want multi-labels as an output, so we use a binary crossentropy loss function along with the sigmoid activation function. Its output dimension is equal to the number of categories available.

- CNN: The CNN model consists of a spatial dropout layer, followed by one CNN layer with kernel size of 3, and 1024 filters, followed by a global max pool layer and a dropout layer.
- RNN: We used both LSTM and GRU models. The LSTM model consists of three layers, while the GRU model consists of four layers. These selections have been carefully chosen by testing different implementations of these models, we then picked the methods that were giving best accuracies. Both models are improved versions of standard RNN, which solve the vanishing gradient problem.
- BIRNN: For an enhanced performance on sequence classification problems, both RNN models are wrapped by a Bidirectional wrapper. The data are fed to the learning algorithm once from the beginning to the end, and once in reverse. Running the input bidirectionally will result in the network understanding the context better. In the forward run, both LSTM and GRU preserve information from the past, and in the backward run, future information is preserved.
- Bi-LSLTM - CRF: Adding a CRF layer to a bidirectional neural network has been proven to be very effective for sequence labeling tasks. Since the CRF model is a unidirectional model, providing the sequence in both directions will decrease the ambiguity of the words in the sequence.
- CNN + RNN: We generate a new CRNN architecture by combining each of CNN and RNN layers together. The new architecture applies a spatial dropout for the embedding layer, we then have a Conv1D layer followed by an RNN layer, we then add a global max pooling layer followed by a dropout layer to apply regularization. Finally, we have a dense layer followed by the output layer.
- HAN: Applying an additional layer after the RNNs models, called attention layer. Along with solving the long-term memory issues with RNNs, the output sequence generated will be conditional on selective items in the input sequence. The hierarchical attention network is extremely good at predicting the class of a given document, because it recognizes its structure. A document is constructed by first encoding the words, then encoding the sentences to form a document representation. An attention mechanism is also used in a HAN, to make sure that the context of a word or a sentence is accounted for. Words and sentences also differ in importance in regards to the main message of the document.

6 Experimental results and discussion

6.1 Single-label classification

The objective is to put 11 classifying models to the test and determine how successful they are to single-label Arabic news articles. We will perform single-label classification on a subset of our single-labeled dataset. After that we will compare the performance of the same models on a recently reported Arabic dataset ‘Akhbarona’ [25, 26] that contains seven categories in total. In this experiment, we used the 80% for training, 20% for testing split. The training set consists of 71,707 articles, while the testing set contained 17,432 articles.

We calculate the accuracy score to evaluate the performance of the classifiers. The accuracy score is the total number of correctly classified samples over the total number of samples. From the training set, we extracted about 344 k features. Additionally, pre-processing of input documents is performed in order to remove all the non-Arabic characters. When dealing with textual data collect from the web, it is highly advised to use this method. We proceed further by cleaning the scraped articles and erasing all the elongation, digits, punctuation, isolated chars, Latin letters, Qur’anic symbols, and other marks that were possibly included.

We believe that applying normalization on the collected text is not a necessary step, even though majority of research works on Arabic NLP tasks do implement normalization. There are enough samples provided in the dataset to represent Arabic character-set. It is worth noting that in some cases, the normalization step can change the semantics of some words. Normalization, which is a widely adopted practice in Arabic computational linguistics, is the process of unifying the orthography of some Arabic characters. Namely, alif forms [ا ، أ ، إ ، آ] to [ا] , hamza forms [ئ، ؤ، ء] to [ء] , haa/taa marbootah [ه، ة] to [ه] , and yaa/alif maqsura [ى، ي] to [ى] . The normalization step is meant to reduce the vocabulary space. However, this process may lead to losing some key features as the meaning of some words would change after normalization. For example, the word “فأر” (means “mouse”) while “فار” after normalization means ’escaped’ or ’كرة,’, which means ’football,’ after normalization becomes “كره” , which means “hatred”. Such meaning-change could result in dropping some important features. In addition, with a large corpus such as the ones proposed in this work, there is no need for this pre-processing step. In fact, the results show that the non-normalized word representation is not seriously hampered by the lack of text normalization.

To implement the classifiers, we used Scikit-learn and kept using the default hyper-parameters in addition to $L1$

penalty for some of the models. We used the testing set to test the proposed classifiers. The high accuracy scores indicate how robust the proposed system is based on the tuned hyper-parameters.

Figure 4 demonstrates the accuracy scores that were obtained by the classifiers. 94.8% was the average of the accuracy score. The best result, 97.9%, was achieved by the SVM classifier. On the other hand, the worst result, which is 87.7%, was produced by the AdaBoost classifier. Furthermore, close results that range from 97.5% to 97.9% were produced by four classifiers.

The two classifiers, MultinomialNB and KNeighbors, produced the accuracies of 96.3% and 95.4%, respectively, meanwhile the remaining classifiers produced scores that range from 87.7% to 94.4%, which is below the average compared to the previous classifiers. Figures 5 displays the confusion matrix for the best and worst classifiers, which are SVM and AdaBoost, respectively.

Furthermore, Table 4 states that SVM scored the highest F1-score 98%, similar to the majority voting classifier, while the lowest score of 88% was produced by the AdaBoost classifier. We also include ROC, Hamming score, F1-score, precision, and recall metrics. We demonstrate the robustness of the best classifier, SVM, by showing the results of prediction. Figure 6 shows an article, from the testing set, which is originally tagged as ‘Technology’. SVM model was able to classify the article under the same category, with an accuracy of 95.7%. We examined a sample of the incorrectly classified documents from the testing dataset in order to recognize the reason behind the misclassification. We realized that several documents can be argued as incorrectly classified. Figure 7 shows an article that was initially classified under ‘Technology.’ After reading the article, we concluded that it belongs to the ‘Business’ class as well, which is in line with the prediction of SVM. This is a strong indicator of the robustness of the SVM classifier. In fact, this motivates the need for multi-label text classification as single-label classification is insufficient.

To further verify the performance, we tested the classifiers on ’Akhbarona,’ [25], which is an unbalanced dataset that includes 46,900 articles in total. This dataset has 7 main classes, which are ’Sports,’ ’Politics,’ ’Medicine,’ ’Religion,’ ’Business,’ ’Technology’ and ’Culture.’ Pre-processing to remove elongation, punctuation, digits, single characters, Quranic symbols, Latin letters, and other marks is initially performed on the dataset. Similar to our earlier training and testing, this dataset was also split into 80% articles for training and 20% for testing. It was safe to expect that the accuracy scores will be lower for two reasons. The first is due to the unbalanced dataset that would cause the classifiers to be biased toward a specific category. Second, when the original number of classes increases, the probability of incorrectly classifying a document increases as well.

Table 5 displays a summary of the accuracy results on the Akhbarona dataset. The best result, 94.4%, was produced by the SVM classifier. However, the Adaboost gave the worst result of 77.9%. In addition, four out of 11 classifiers report accuracy scores that range between 93.9% and 94.4%. As for the remaining 7 classifiers, only the KNeighbors classifier produced an accuracy of 90.8%, which is higher than the average. The other 6 classifiers produced results that vary from 77.9% to 88.4%. The table also shows the rest of the evaluation metrics: precision, recall, and F1-score.

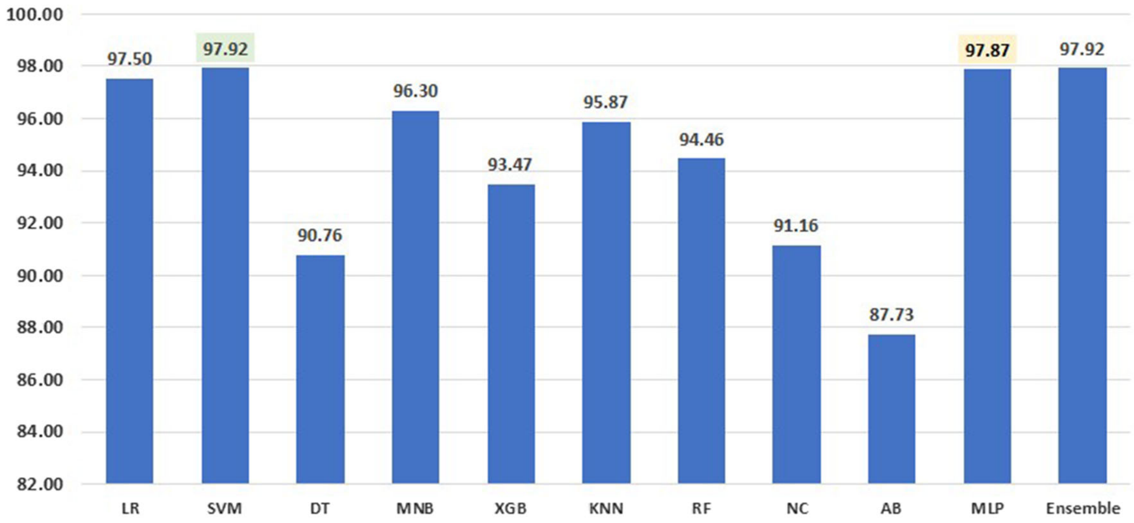


Fig. 4 Performance of all classifiers on our proposed dataset

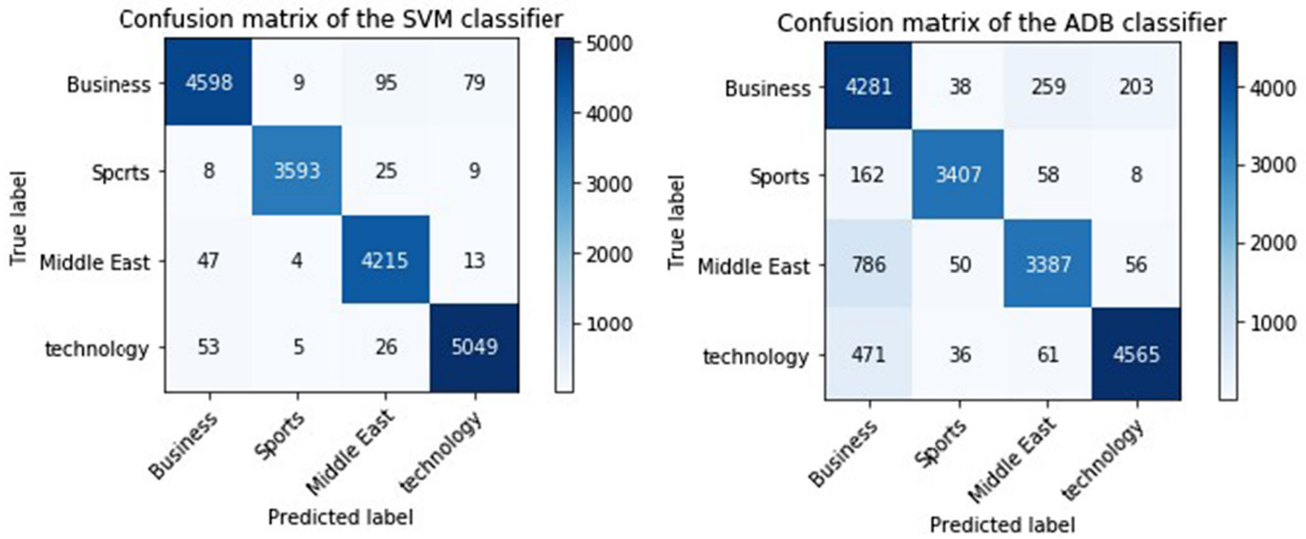


Fig. 5 The confusion matrices for highest(SVM) and lowest (AD) performers

Table 4 Evaluation metrics for all classifiers on the testing dataset

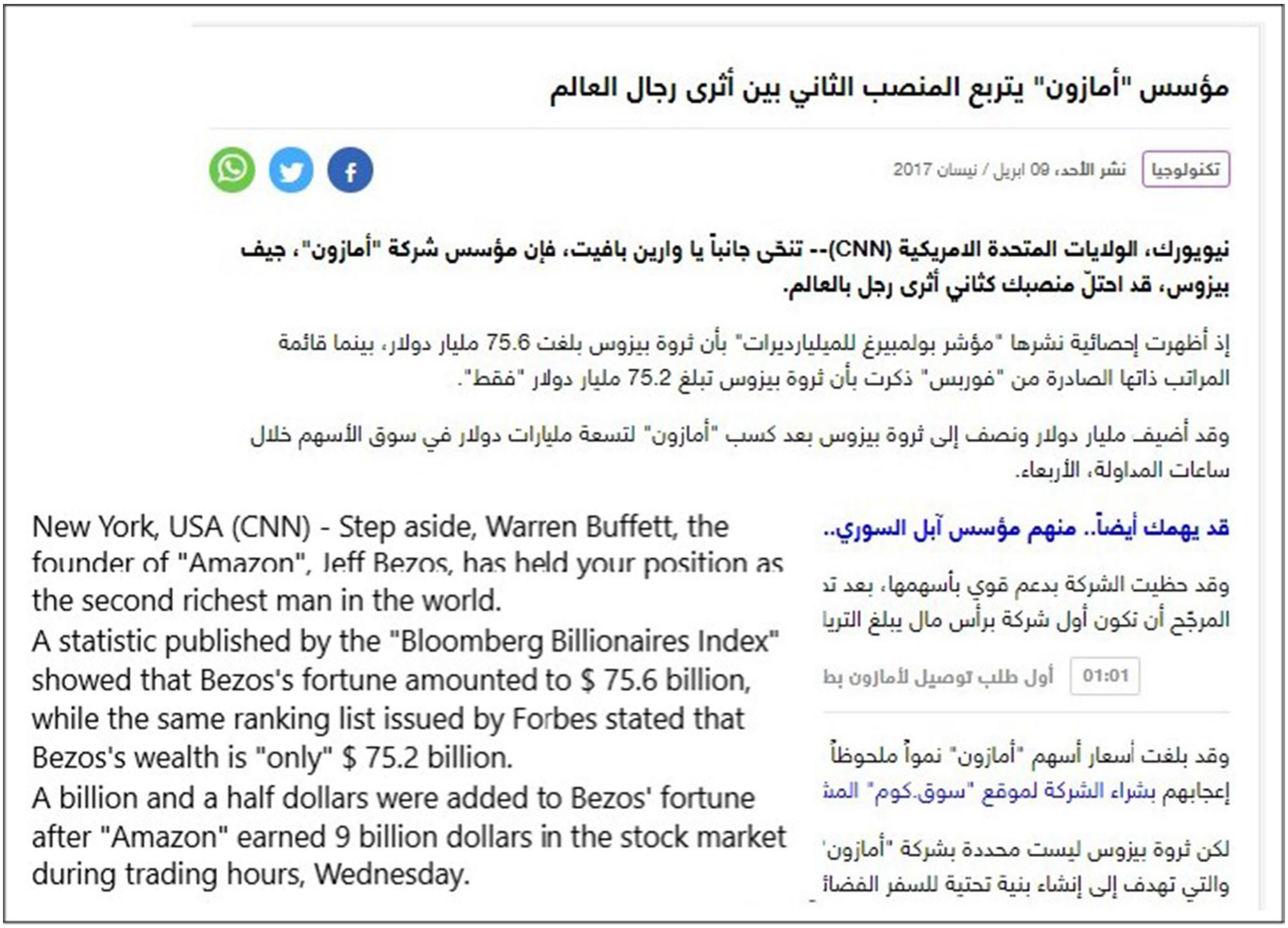
Classifier	Evaluation metrics					
	Accuracy	Ham. loss	F1 Score	Precision	Recall	ROC
LR	97.50	0.025	97.57	97.58	97.57	99.84
SVM	97.92	0.021	97.98	97.99	97.98	99.87
DT	90.76	0.092	90.92	90.94	90.92	93.90
MNB	96.30	0.037	96.37	96.50	96.31	99.76
XGB	93.47	0.065	93.64	93.89	93.51	99.27
KNN	95.87	0.041	95.95	95.97	95.94	99.14
RF	94.46	0.055	94.61	94.64	94.60	99.21
NC	91.16	0.088	91.32	92.53	91.37	92.40
ADB	87.73	0.123	88.15	89.01	87.84	94.08
MLP	**97.87**	**0.021**	97.93	97.93	97.92	**99.88**
Ensemble	97.92	0.021	97.98	97.99	97.98	99.88

6.2 Multi-label classification

The aim, in this section, is to investigate the performance of classifying Arabic news articles with multiple labels. We experiment with classical classifiers and with deep learning. The total number of articles in the subset used in these experiments is 148,376 articles. We chose the labels with the highest frequency, because the performance of supervised classifiers is highly dependent on the number of instances for each label. Figure 8 shows the count of the 21 labels chosen from the dataset.

For the classical classifiers approach, we split the dataset into 80% training set consisting of 118,700 labeled articles, and 20% testing set consisting of 29,676 articles. For the deep learning approach, we split the dataset into 80% training set, containing 118,700 articles (where 10% of them were used for validation 11,870), and 20% into the testing set containing 29,676. The validating dataset is used to tune some of the model's hyperparameters such as: (layer size, hidden unit number and regularization term). The same text pre-processing steps used on the single-labeled dataset are used on the multi-labeling dataset.

We implemented and tested the proposed classifiers on the multi-labeled dataset. We used a custom accuracy metric to evaluate the accuracy of the predictions. It is the ratio of correctly predicted tags (output as 1) over total expected tags (originally 1 in dataset). The more correct labels the model predicts, the more accurate it is. We chose a threshold of 50%, meaning we consider the labels with a probability percentage higher or equal to 50% to be correct. The second evaluating metric used is hamming loss. Hamming loss metric is often used to evaluate the performance of multi-labeling classifiers. It is the fraction of wrong predicted labels to the total number of labels. The



مؤسس "أمازون" يتربع المنصب الثاني بين أثرى رجال العالم

تكنولوجيا نشر الأحد، 09 ابريل / نيسان 2017

نيويورك، الولايات المتحدة الامريكية (CNN)-- تنحّى جانباً يا وارين بافيت، فإن مؤسس شركة "أمازون"، جيف بيزوس، قد احتلّ منصبك كثاني أثرى رجل بالعالم.

إذ أظهرت إحصائية نشرها "مؤشر بولمبيرغ للمليارديرات" بأن ثروة بيزوس بلغت 75.6 مليار دولار، بينما قائمة المراتب ذاتها الصادرة من "فوربس" ذكرت بأن ثروة بيزوس تبلغ 75.2 مليار دولار "فقط".

وقد أضيف مليار دولار ونصف إلى ثروة بيزوس بعد كسب "أمازون" لتسعة مليارات دولار في سوق الأسهم خلال ساعات المداولة، الأربعاء.

New York, USA (CNN) - Step aside, Warren Buffett, the founder of "Amazon", Jeff Bezos, has held your position as the second richest man in the world.
A statistic published by the "Bloomberg Billionaires Index" showed that Bezos's fortune amounted to $ 75.6 billion, while the same ranking list issued by Forbes stated that Bezos's wealth is "only" $ 75.2 billion.
A billion and a half dollars were added to Bezos' fortune after "Amazon" earned 9 billion dollars in the stock market during trading hours, Wednesday.

قد يهمك أيضاً.. منهم مؤسس آبل السوري..

وقد حظيت الشركة بدعم قوي بأسهمها، بعد تد
المرجّح أن تكون أول شركة برأس مال يبلغ التريا

01:01 أول طلب توصيل لأمازون بط

وقد بلغت أسعار أسهم "أمازون" نمواً ملحوظاً
إعجابهم بشراء الشركة لموقع "سوق.كوم" المش

لكن ثروة بيزوس ليست محددة بشركة "أمازون"
والتي تهدف إلى إنشاء بنية تحتية للسفر الفضا

Fig. 6 A correctly tagged news-article as ’Technology’

smaller the value of hamming loss, the better results the model is achieving.

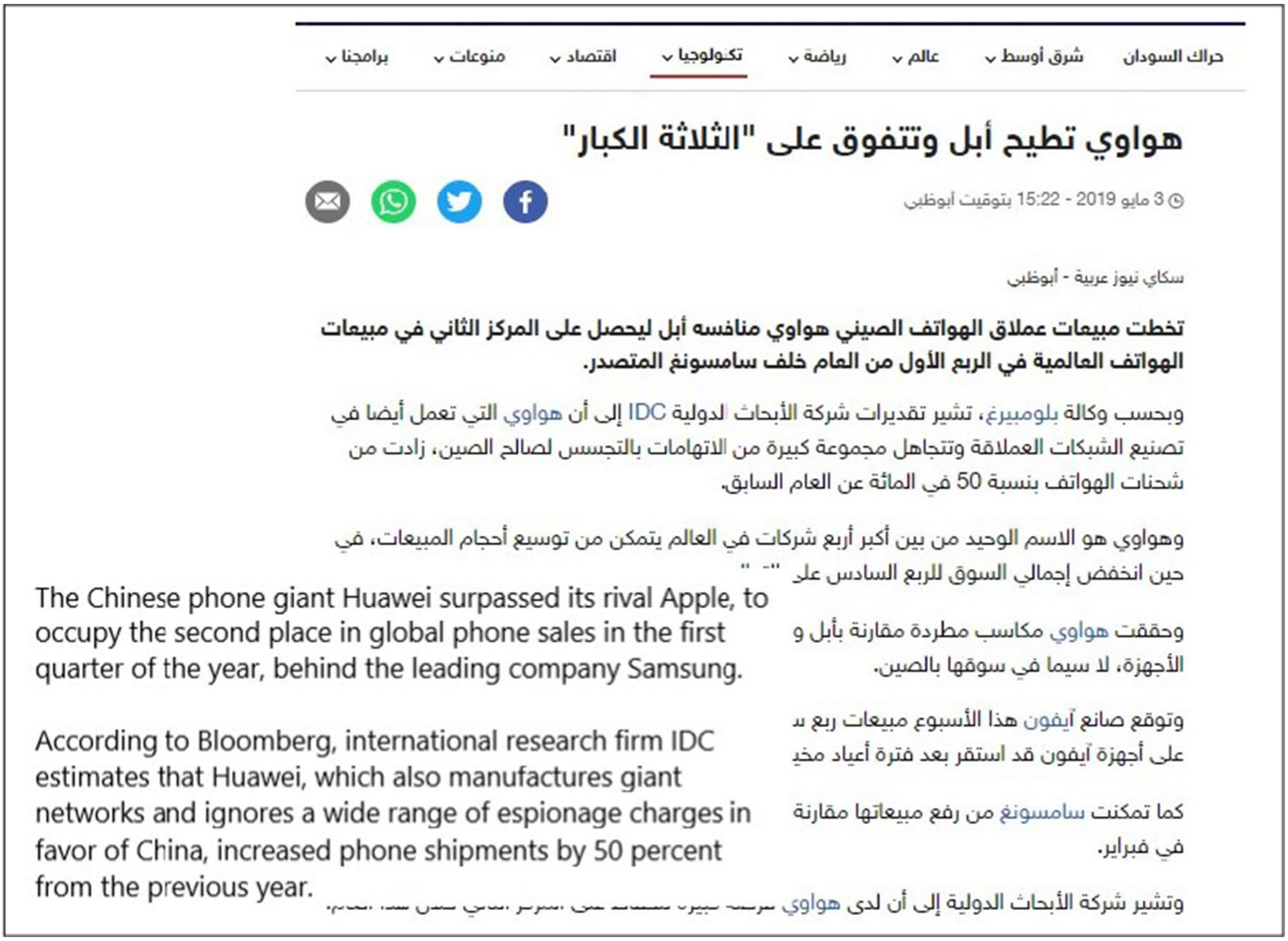
Table 6 displays the accuracy scores achieved by two shallow-learning (SL) classifiers. The XGBoost classifier scored the highest accuracy of 84.73%, while the Logistic Regression scored the lowest accuracy of 81.34%. The hamming loss, which calculates the ratio of wrongly predicted labels to the total number of labels. The lower the percentage is the better. Both classifiers had comparable hamming loss scores, but the XGBoost scored the lowest, achieving 2.24%. We also include ROC, Hamming score, F1-score, precision, and recall metrics.

We train-test-validate ten deep neural (DL) networks, with different architectures, seeking the best at performing the task at hand. Figure 9 displays the resulting accuracy percentages, using the same accuracy metric described earlier. It shows that the CNN-GRU scored 94.85% and surpassed all the other classifiers, including the SL classifiers for multi-labels. Table 7 shows all metrics for each classifier including Hamming score, F1-score, precision, recall, and ROC. The CGRU achieved the best Hamming, F1, and recall scores. However, GRU reported the best precision score and LSTM reported the best ROC value. It is notable that the scores of GRU, BILSTM, and CGRU are close.

Focusing more on the CNN-GRU model, Table 8 shows the precision, recall and f1 scores of the 21 multi-labels. Table 10 displays the average scores of the model with respect to micro, macro, weighted, and samples averages for the CGRU classifier (Table 9).

Figure 10 shows how the relationships between the 21 true labels are present in the testing set. The edges in this graph present the instances in which the two labels at the end of each edge are present. The width of the edge indicated the number of instances. Figure 11 shows the relationships produced by the model’s predictions. The way the labels are appearing together is very different than how they were appearing in the collected dataset. The model is classifying the articles under different main categories. It has learned efficiently enough to start identifying multiple topics in the articles.

Figure 12 displays an example of a news article classified by the model. Originally, it is tagged as ’Business’. The article discusses the wealth of ’ARAMCO’ and how it



هواوي تطيح أبل وتتفوق على "الثلاثة الكبار"

3 مايو 2019 - 15:22 بتوقيت أبوظبي

سكاي نيوز عربية - أبوظبي

تخطت مبيعات عملاق الهواتف الصيني هواوي منافسه أبل ليحصل على المركز الثاني في مبيعات الهواتف العالمية في الربع الأول من العام خلف سامسونغ المتصدر.

وبحسب وكالة بلومبيرغ، تشير تقديرات شركة الأبحاث لدولية IDC إلى أن هواوي التي تعمل أيضا في تصنيع الشبكات العملاقة وتتجاهل مجموعة كبيرة من الاتهامات بالتجسس لصالح الصين، زادت من شحنات الهواتف بنسبة 50 في المائة عن العام السابق.

وهواوي هو الاسم الوحيد من بين أكبر أربع شركات في العالم يتمكن من توسيع أحجام المبيعات، في حين انخفض إجمالي السوق للربع السادس علـ

وحققت هواوي مكاسب مطردة مقارنة بأبل و
الأجهزة، لا سيما في سوقها بالصين.

وتوقع صانع آيفون هذا الأسبوع مبيعات ربع س
على أجهزة آيفون قد استقر بعد فترة أعياد مخي

كما تمكنت سامسونغ من رفع مبيعاتها مقارنة
في فبراير.

وتشير شركة الأبحاث الدولية إلى أن لدى هواوي

The Chinese phone giant Huawei surpassed its rival Apple, to occupy the second place in global phone sales in the first quarter of the year, behind the leading company Samsung.

According to Bloomberg, international research firm IDC estimates that Huawei, which also manufactures giant networks and ignores a wide range of espionage charges in favor of China, increased phone shipments by 50 percent from the previous year.

Fig. 7 A misclassified news-article as ’Business’; Originally it is tagged as ’Technology’

Table 5 Classifiers accuracy scores on ’Akhbarona’ dataset

Classifiers	Evaluation metrics			
	Accuracy	Precision	Recall	F1-score
LR	93.9	0.94	0.94	0.94
SVM	**94.4**	0.94	0.94	0.94
DT	83.0	0.83	0.83	0.83
MNB	88.0	0.91	0.88	0.88
XGB	88.4	0.89	0.88	0.88
KNN	90.8	0.91	0.91	0.91
RF	87.8	0.88	0.88	0.88
NC	86.2	0.89	0.86	0.87
ADB	**77.9**	0.80	0.78	0.78
MLP	94.1	0.94	0.94	0.94
Ensemble	94.3	0.94	0.94	0.94

is beating both ’google’ and ’apple’. Upon feeding the article to the CGRU model, the predicted labels are ’Technology,’ ’Business,’ ’Saudi Business,’ ’Google,’ and ’Apple.’ This is an accurate and sufficient tagging of the article. The English translation of Fig. 12 is depicted as well.

Figure 13, is another article classified under ‘Barca’ in the website. The resulting tags from CGRU are ’Sports’ and ’Real Madrid’. If you carefully read this article, you will easily find that the article has nothing to do with ’Barca’. On the contrary, the article is mainly talking about ’Real Madrid’ and the injury of its player ’Hazard’. This example shows how CNN model is outperforming the original tagging of the article. This example confirms the fact that such tagging models may solve the issue of re-occurring human tagging errors. The English translation of Fig. 13 is depicted as well.

To complete the analyses, we discuss the upper bounds of the computational cost of the implemented classifiers. We use the following notation: n is the number of training samples, f is the number of features, n_t is the number of trees with depth d (DT and similar classifiers), k is the number of neighbors (KNN), n_v is the number of support

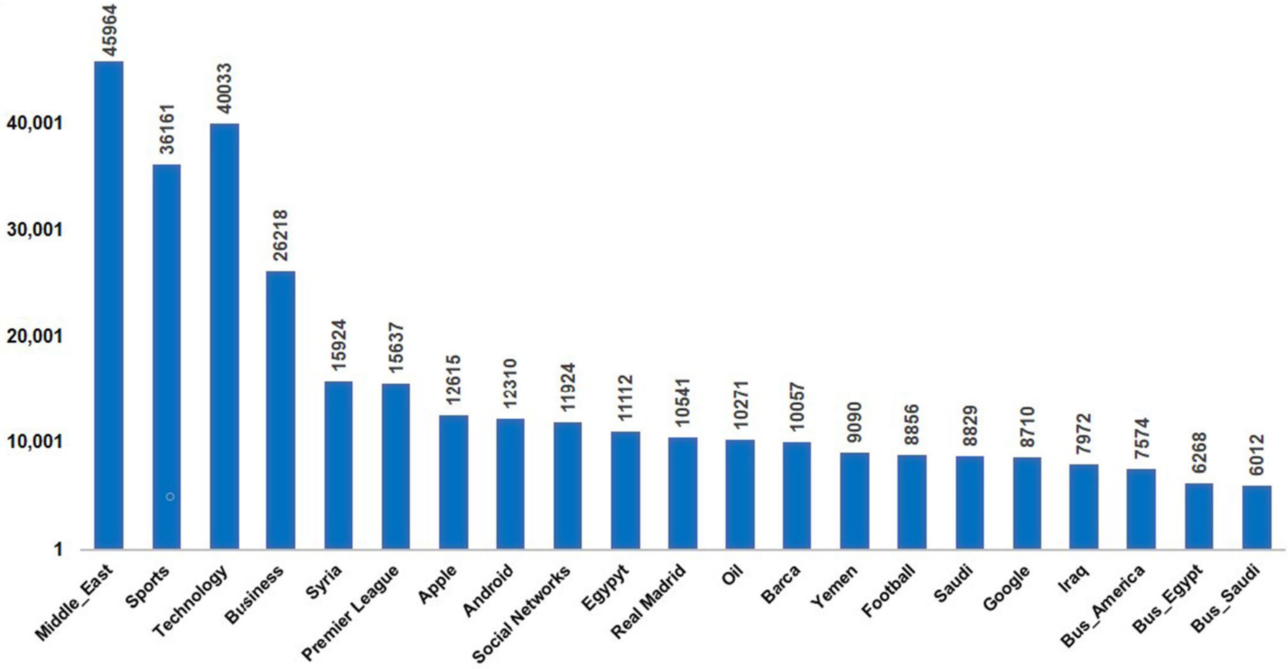


Fig. 8 Count of the labels used in CGRU experiment

Table 6 Evaluation metrics of the SL classifiers for the multi-label classification task

Classifier	Evaluation metrics					
	Accuracy	Ham. loss	F1 Score	Precision	Recall	ROC
OVR-LR	81.34	2.50	75.62	88.67	69.01	98.40
OVR-XGB	84.73	2.24	78.86	87.59	74.87	98.47

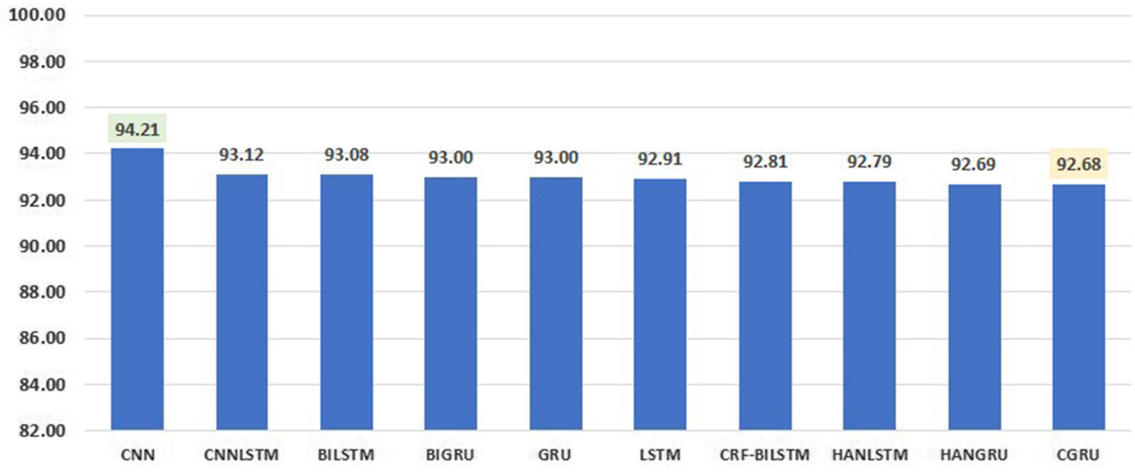


Fig. 9 Accuracy scores for all deep neural networks

vectors (SVM), and n_i is the number of neurons at layer i in a neural network with l layers that is trained e epochs. Approximate upper bounds for training time as well as prediction time are listed in Table 10. It should be noted that training time is minor as it is carried out once and offline. However, prediction time is more important as it is used for every prediction after training the classifiers.

To verify Table 10, we measured the time consumed for training the models, we ran an experiment with n varies between[1000,5000,10000,20000] and f takes the values

Table 7 Evaluation metrics for all deep learning classifiers for the multi-label classification task

Classifier	Evaluation metrics					
	Accuracy	Ham. loss	F1 Score	Precision	Recall	ROC
CNN	91.34	1.61	89.13	90.99	87.51	95.08
CNNLSTM	91.34	1.61	89.13	90.99	87.51	95.08
BILSTM	94.03	1.27	90.25	92.31	88.48	97.06
BIGRU	91.34	1.61	89.13	90.99	87.51	95.08
GRU	94.28	**1.21**	90.55	**92.90**	88.70	98.04
LSTM	90.17	1.78	86.85	90.61	83.92	**98.70**
CRF-BILSTM	91.34	1.61	89.13	90.99	87.51	95.08
HANLSTM	92.92	1.45	90.60	91.05	90.57	83.83
HANGRU	92.96	1.43	90.66	91.16	90.40	93.52
CGRU	**94.85**	**1.21**	**90.72**	92.06	**89.74**	97.74

Table 8 Evaluation metrics of the CNN-GRU classifier per each of the 21 labels

Label	Precision	Recall	F1-score
Business	97.81	97.95	97.88
oil	90.58	94.27	92.39
Business America	89.12	76.30	82.21
Business Egypt	93.24	88.91	91.02
Business SA	81.23	87.18	84.10
ME	99.16	98.87	99.01
Syria	94.72	92.99	93.84
Egypt	91.84	93.57	92.69
Yaman	95.91	88.47	92.04
Saudi Arabia	89.94	79.47	84.38
Iraq	94.31	89.28	91.73
Sports	99.67	99.88	99.77
Premier League	88.84	95.73	92.16
Real Madrid	90.53	90.78	90.65
Barca	89.47	90.79	90.13
Football	82.62	57.37	67.72
Tech	99.52	99.78	99.65
Android	89.49	89.45	89.47
Apple	93.11	89.44	91.24
Google	87.84	88.86	88.35
Social Media	94.19	95.31	94.75

Table 9 Average evaluation scores of the CNN-GRU classifier

CNNGRU	Precision	Recall	F1-score
Micro avg	94.72	93.48	94.09
Macro avg	92.06	89.74	90.72
Weighted avg	94.63	93.48	93.95
Samples avg	95.67	94.85	94.69

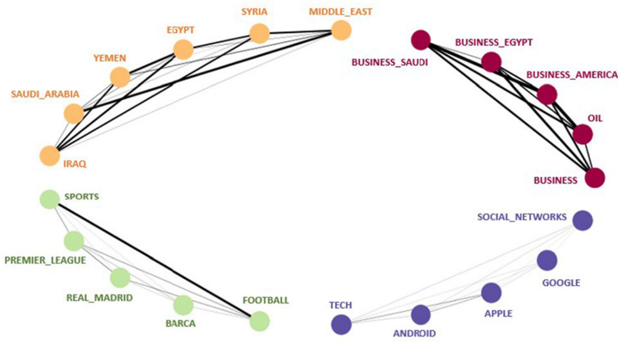


Fig. 10 Relationships of true labels in the testing dataset

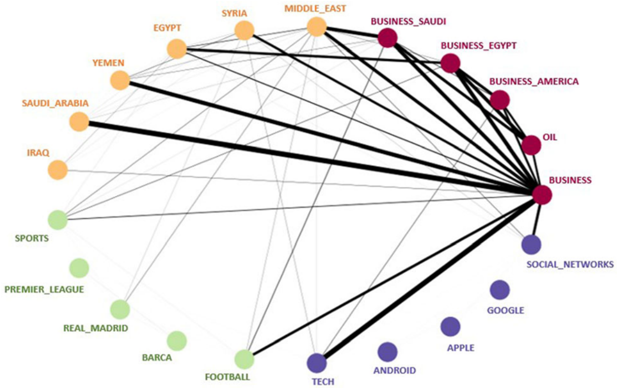
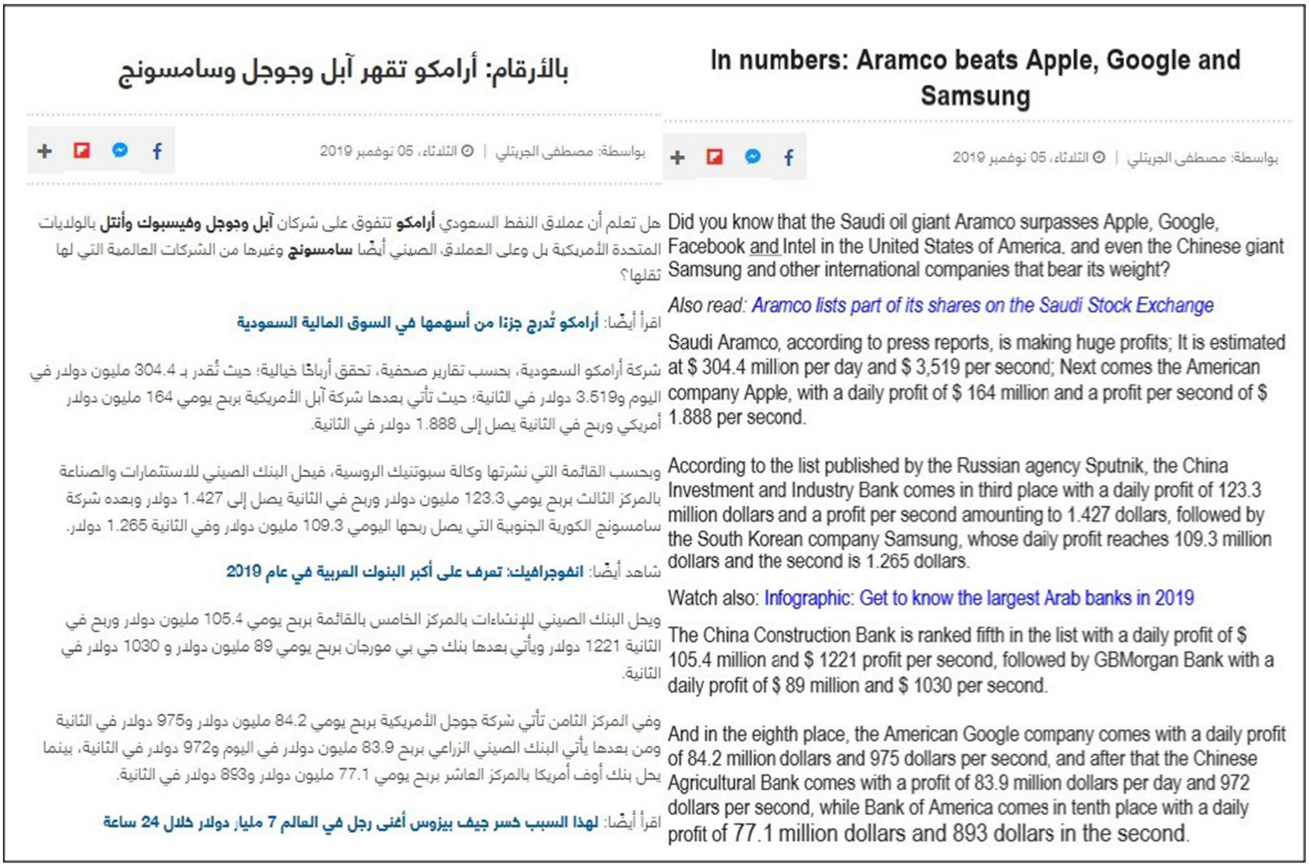


Fig. 11 Relationships of predicted labels by CNN

[100,500,1000]. The average training time in seconds is depicted in Fig. 14.

To predict the execution time during training for a CNN model, we need to consider the features that contribute to this time estimate, [35]. However, such features are numerous and include layer features, convolution and pooling features, and hardware features. Such features can easily vary tremendously based on the deep learning network being implemented. Layer features may include activation function, optimizer, and batch size. Of course, a



بالأرقام: أرامكو تقهر آبل وجوجل وسامسونج

بواسطة: مصطفى الجريتلي | الثلاثاء، 05 نوفمبر 2019

هل تعلم أن عملاق النفط السعودي **أرامكو** تتفوق على شركان **آبل وجوجل وفيسبوك وأنتل** بالولايات
المتحدة الأمريكية بل وعلى العملاق الصيني أيضًا **سامسونج** وغيرها من الشركات العالمية التي لها
ثقلها؟

اقرأ أيضًا: **أرامكو تُدرج جزءًا من أسهمها في السوق المالية السعودية**

شركة أرامكو السعودية، بحسب تقارير صحفية، تحقق أرباحًا خيالية؛ حيث تُقدر بـ 304.4 مليون دولار في
اليوم و3.519 دولار في الثانية؛ حيث تأتي بعدها شركة آبل الأمريكية بربح يومي 164 مليون دولار
أمريكي وربح في الثانية يصل إلى 1.888 دولار في الثانية.

وبحسب القائمة التي نشرتها وكالة سبوتنيك الروسية، فيحل البنك الصيني للاستثمارات والصناعة
بالمركز الثالث بربح يومي 123.3 مليون دولار وربح في الثانية يصل إلى 1.427 دولار وبعده شركة
سامسونج الكورية الجنوبية التي يصل ربحها اليومي 109.3 مليون دولار وفي الثانية 1.265 دولار.

شاهد أيضًا: **انفوجرافيك: تعرف على أكبر البنوك العربية في عام 2019**

ويحل البنك الصيني للإنشاءات بالمركز الخامس بالقائمة بربح يومي 105.4 مليون دولار وربح في
الثانية 1221 دولار ويأتي بعدها بنك جي بي مورجان بربح يومي 89 مليون دولار و 1030 دولار في
الثانية.

وفي المركز الثامن تأتي شركة جوجل الأمريكية بربح يومي 84.2 مليون دولار و975 دولار في الثانية
ومن بعدها يأتي البنك الصيني الزراعي بربح 83.9 مليون دولار في اليوم و972 دولار في الثانية، بينما
يحل بنك أوف أمريكا بالمركز العاشر بربح يومي 77.1 مليون دولار و893 دولار في الثانية.

اقرأ أيضًا: **لهذا السبب كسر جيف بيزوس أغنى رجل في العالم 7 مليار دولار خلال 24 ساعة**

In numbers: Aramco beats Apple, Google and Samsung

بواسطة: مصطفى الجريتلي | الثلاثاء، 05 نوفمبر 2019

Did you know that the Saudi oil giant Aramco surpasses Apple, Google, Facebook and Intel in the United States of America, and even the Chinese giant Samsung and other international companies that bear its weight?

Also read: Aramco lists part of its shares on the Saudi Stock Exchange

Saudi Aramco, according to press reports, is making huge profits; It is estimated at $ 304.4 million per day and $ 3,519 per second; Next comes the American company Apple, with a daily profit of $ 164 million and a profit per second of $ 1.888 per second.

According to the list published by the Russian agency Sputnik, the China Investment and Industry Bank comes in third place with a daily profit of 123.3 million dollars and a profit per second amounting to 1.427 dollars, followed by the South Korean company Samsung, whose daily profit reaches 109.3 million dollars and the second is 1.265 dollars.

Watch also: Infographic: Get to know the largest Arab banks in 2019

The China Construction Bank is ranked fifth in the list with a daily profit of $ 105.4 million and $ 1221 profit per second, followed by GBMorgan Bank with a daily profit of $ 89 million and $ 1030 per second.

And in the eighth place, the American Google company comes with a daily profit of 84.2 million dollars and 975 dollars per second, and after that the Chinese Agricultural Bank comes with a profit of 83.9 million dollars per day and 972 dollars per second, while Bank of America comes in tenth place with a daily profit of 77.1 million dollars and 893 dollars in the second.

Fig. 12 An example of a news article correctly tagged, with 5 tags, by the CGRU model

layer in the network may have far more of these features. Convolutional and pooling features include matrix size, input depth and padding, output depth, stride size, and kernel size. Hardware features are those related to GPU/CPU technology used, GPU/CPU count, memory, and clock speed.

In order to work out a prediction model of the computational time cost for training a CNN network that has multiple layers, we need to compute the time required for a forward and backward pass on a single batch for a single epoch. Then, the total time-cost estimate (T) for training the CNN network is:

$$T = e\,b \sum_{i=0}^{l} t_i$$

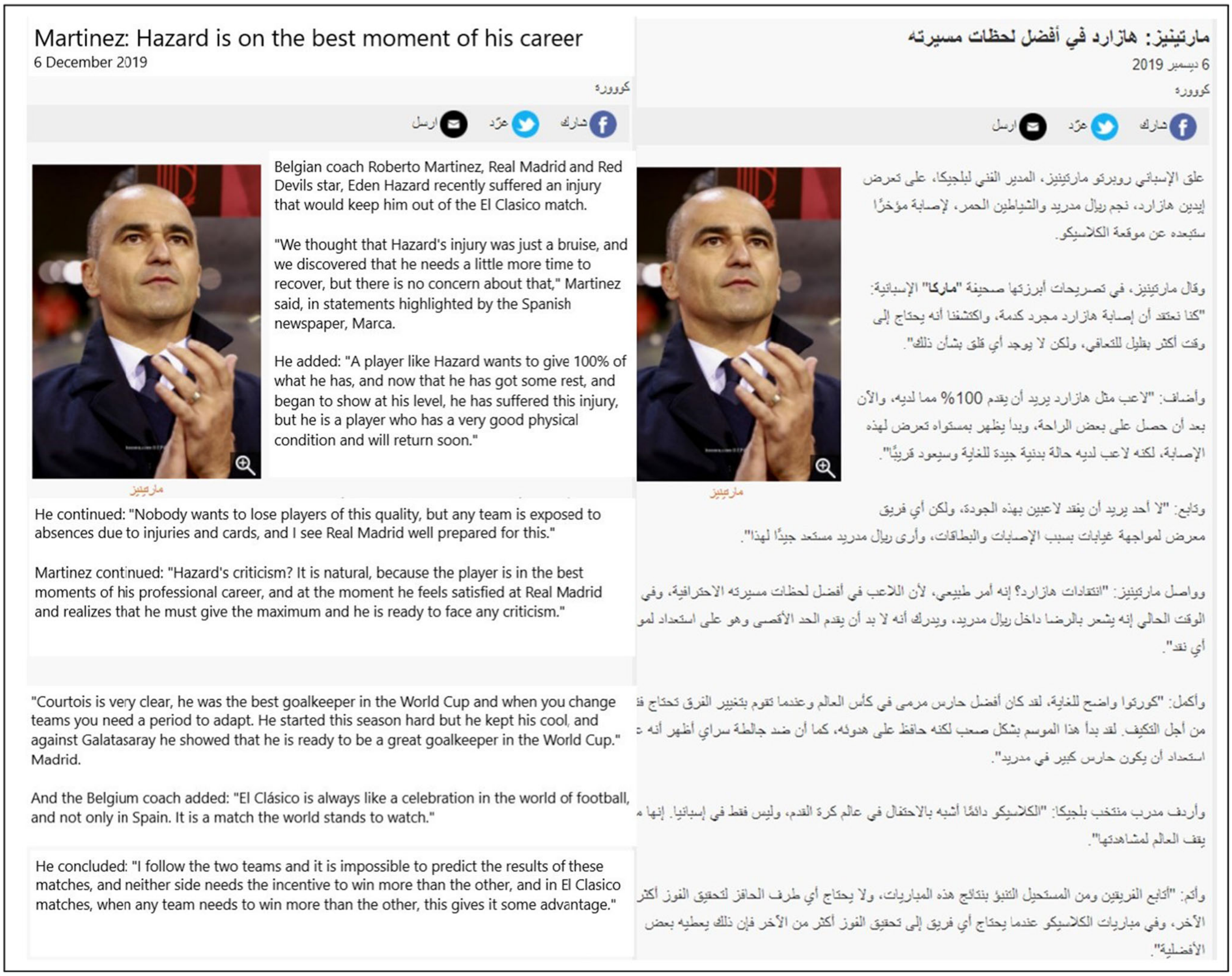
where l is the number of layers in the CNN model, t_i is the batch execution time estimate for layer i, b is the number of batches, and e is the number of epochs. Of course, e is a constant number that can be ignored.

7 Conclusions

We have presented an automatic general text categorization system for Arabic text that can handle both single-label as well as multi-label tagging tasks. We described a single-labeled dataset (90k Arabic documents) with their labels collected from seven different news portals. In addition, we reported a multi-labeled dataset comprising 293k Arabic articles, with their tags, collected from 10 news portals.

We examined the first dataset by implementing 12 shallow-learning classifiers for the single-label text classification task. Although the SVM model outperformed the rest, the final accuracy scores confirm the robustness of all classifiers, ranging between 87% and 97%. We also used the voting classifier, in pursuit of better accuracy, using an ensemble model. However, its resulting performance is analogous to SVM.

The second dataset was examined by implementing ten different deep learning neural networks along with two shallow learning classifiers for the multi-labeling task. A custom accuracy metric was implemented to evaluate the performance of the developed models. For the shallow learning case, the OVR-XGBoost classifier reported the higher accuracy than the OVR-Logistic Regression



Martinez: Hazard is on the best moment of his career
6 December 2019
كووورة

Belgian coach Roberto Martinez, Real Madrid and Red Devils star, Eden Hazard recently suffered an injury that would keep him out of the El Clasico match.

"We thought that Hazard's injury was just a bruise, and we discovered that he needs a little more time to recover, but there is no concern about that," Martinez said, in statements highlighted by the Spanish newspaper, Marca.

He added: "A player like Hazard wants to give 100% of what he has, and now that he has got some rest, and began to show at his level, he has suffered this injury, but he is a player who has a very good physical condition and will return soon."

He continued: "Nobody wants to lose players of this quality, but any team is exposed to absences due to injuries and cards, and I see Real Madrid well prepared for this."

Martinez continued: "Hazard's criticism? It is natural, because the player is in the best moments of his professional career, and at the moment he feels satisfied at Real Madrid and realizes that he must give the maximum and he is ready to face any criticism."

"Courtois is very clear, he was the best goalkeeper in the World Cup and when you change teams you need a period to adapt. He started this season hard but he kept his cool, and against Galatasaray he showed that he is ready to be a great goalkeeper in the World Cup." Madrid.

And the Belgium coach added: "El Clásico is always like a celebration in the world of football, and not only in Spain. It is a match the world stands to watch."

He concluded: "I follow the two teams and it is impossible to predict the results of these matches, and neither side needs the incentive to win more than the other, and in El Clasico matches, when any team needs to win more than the other, this gives it some advantage."

مارتينيز: هازارد في أفضل لحظات مسيرته
6 ديسمبر 2019
كووورة

علق الإسباني روبرتو مارتينيز، المدير الفني لبلجيكا، على تعرض إيدين هازارد، نجم ريال مدريد والشياطين الحمر، لإصابة مؤخرًا ستبعده عن موقعة الكلاسيكو.

وقال مارتينيز، في تصريحات أبرزتها صحيفة **"ماركا"** الإسبانية: "كنا نعتقد أن إصابة هازارد مجرد كدمة، واكتشفنا أنه يحتاج إلى وقت أكثر بقليل للتعافي، ولكن لا يوجد أي قلق بشأن ذلك".

وأضاف: "لاعب مثل هازارد يريد أن يقدم 100% مما لديه، والآن بعد أن حصل على بعض الراحة، وبدأ يظهر بمستواه تعرض لهذه الإصابة، لكنه لاعب لديه حالة بدنية جيدة للغاية وسيعود قريبًا".

وتابع: "لا أحد يريد أن يفقد لاعبين بهذه الجودة، ولكن أي فريق معرض لمواجهة غيابات بسبب الإصابات والبطاقات، وأرى ريال مدريد مستعد جيدًا لهذا".

وواصل مارتينيز: "انتقادات هازارد؟ إنه أمر طبيعي، لأن اللاعب في أفضل لحظات مسيرته الاحترافية، وفي الوقت الحالي إنه يشعر بالرضا داخل ريال مدريد، ويدرك أنه لا بد أن يقدم الحد الأقصى وهو على استعداد لمو أي نقد".

وأكمل: "كورتوا واضح للغاية، لقد كان أفضل حارس مرمى في كأس العالم وعندما تقوم بتغيير الفرق تحتاج فت من أجل التكيف. لقد بدأ هذا الموسم بشكل صعب لكنه حافظ على هدوئه، كما أن ضد جالطة سراي أظهر أنه ع استعداد أن يكون حارس كبير في مدريد".

وأردف مدرب منتخب بلجيكا: "الكلاسيكو دائمًا أشبه بالاحتفال في عالم كرة القدم، وليس فقط في إسبانيا. إنها م يقف العالم لمشاهدتها".

وأتم: "أتابع الفريقين ومن المستحيل التنبؤ بنتائج هذه المباريات، ولا يحتاج أي طرف الحافز لتحقيق الفوز أكثر الأخر، وفي مباريات الكلاسيكو عندما يحتاج أي فريق إلى تحقيق الفوز أكثر من الأخر فإن ذلك يعطيه بعض الأفضلية".

Fig. 13 An example of a misclassified news article that turns to be good

Table 10 Approximate computational cost for the classifier

Classifier	Training	Prediction
LR	$O(nf)$	$O(f)$
SVM	$O(n^2f + n^3)$	$O(n_vf)$
DT	$O(n^2f)$	$O(df)$
MNB	$O(nf)$	$O(f)$
XGB	$O(nfn_t)$	$O(fn_t)$
KNN	$O(knf)$	$O(kf)$
RF	$O(nlog(n)fn_t)$	$O(dn_t)$
NC	$O(nf$	$O(f)$
ADB	$O(nfn_t)$	$O(fn_t)$
MLP/NN	$O(e(fn_1 + n_1n_2 + \cdot + n_{l-1}n_l))$	$O(fn_1 + n_1n_2 + \cdot + n_{l-1}n_l))$

classifier. Using deep learning models, the accuracy scores were considerably higher than the two shallow-learning classifiers. The highest achieved accuracy score was 94.85% reported by CNN-GRU, while the worst accuracy was 90.17% achieved by LSTM classifier. The rest of the

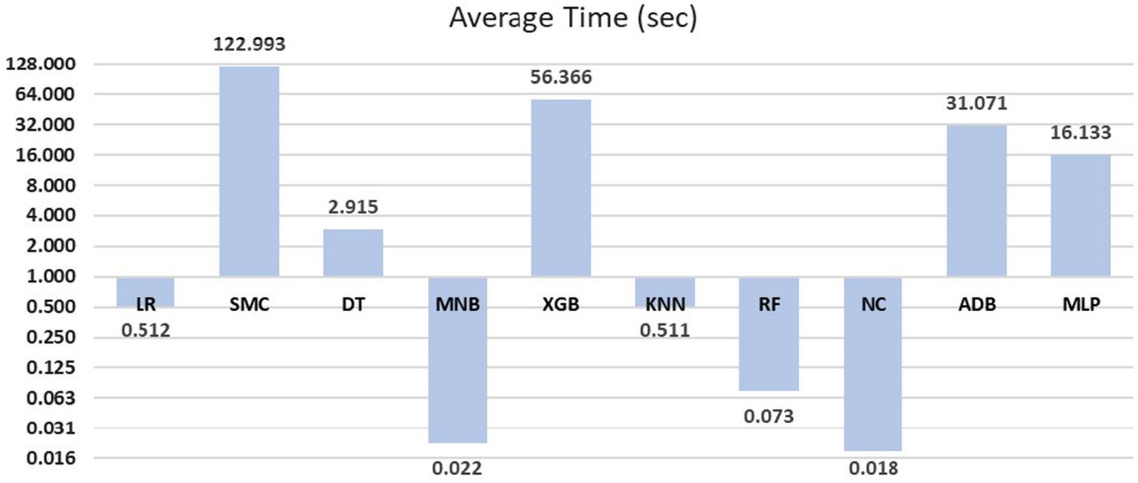


Fig. 14 Average training time for the standard classifiers for multiple *n* samples and *f* features; Logarithmic scale

Table 11 Classifiers parameters

Classifier	parameters settings
LR	C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=100, multi_class='ovr', penalty='l2', solver='lbfgs', tol=0.0001, warm_start=False
SVM	C=1.0, break_ties=False, cache_size=200, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',max_iter=-1, probability=True, shrinking=True, tol=0.001
DT	criterion='gini', min_samples_split=2, min_samples_leaf=1
MNB	alpha=1.0, fit_prior=True
XGB	loss='deviance', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, max_depth=3, warm_start=False, validation_fraction=0.1, tol=0.0001
KNN	n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski
RF	n_estimators=100, *, criterion='gini', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0
NC	metric='euclidean'
ADB	algorithm='SAMME.R', learning_rate=1.0, n_estimators=50
MLP	activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, epsilon=1e-08, hidden_layer_sizes=(100,), learning_rate='constant', learning_rate_init=0.001, max_iter=200, momentum=0.9, n_iter_no_change=10, power_t=0.5, solver='adam', tol=0.0001, validation_fraction=0.1, warm_start=False

models performed very well ranging from 91.34% to 94.28%.

In future, we intend to test different embedding models such as BERT and ELMo besides using transformer networks as well. In addition, we are aiming at increasing the number of labels in the proposed datasets. We intend to make all datasets available to the research community.

Appendix 1

Shallow Learning Classifiers

The different parameters considered for each classifier are detailed in this appendix. We list all parameters required to produce the same results. Unlisted parameters are kept with the default values (Table 11).

Deep Learning Classifiers

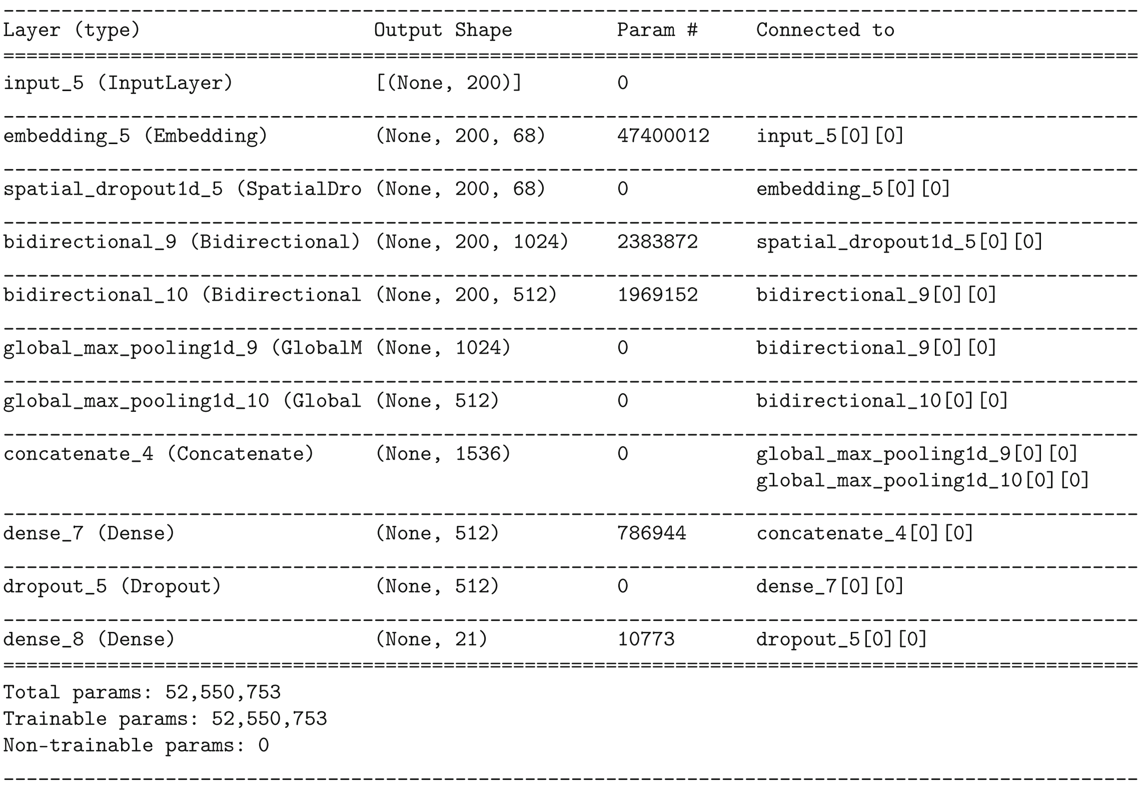
- CNN

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_12 (Embedding)     (None, 200, 68)           47400012
_________________________________________________________________
spatial_dropout1d_12 (Spatia (None, 200, 68)           0
_________________________________________________________________
conv1d_12 (Conv1D)           (None, 200, 1024)         209920
_________________________________________________________________
global_max_pooling1d_12 (Glo (None, 1024)              0
_________________________________________________________________
dense_23 (Dense)             (None, 1024)              1049600
_________________________________________________________________
dropout_12 (Dropout)         (None, 1024)              0
_________________________________________________________________
activation_23 (Activation)   (None, 1024)              0
_________________________________________________________________
dense_24 (Dense)             (None, 21)                21525
_________________________________________________________________
activation_24 (Activation)   (None, 21)                0
=================================================================
Total params: 48,681,057
Trainable params: 48,681,057
Non-trainable params: 0
_________________________________________________________________

Dropout=0.25
Dropout=0.15
epochs=8,
batch_size=128,
validation_split=0.1,
_________________________________________________________________
```

- CNNLSTM

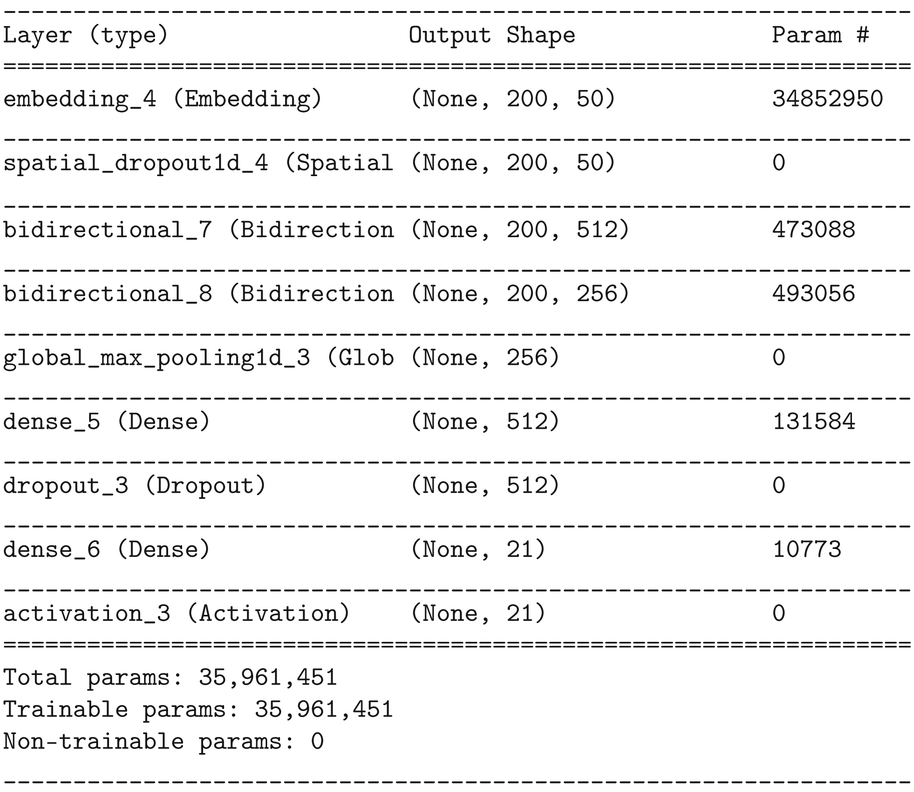
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 200, 50)           34852950
_________________________________________________________________
spatial_dropout1d_3 (Spatial (None, 200, 50)           0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 200, 512)          77312
_________________________________________________________________
cu_dnnlstm_5 (CuDNNLSTM)     (None, 200, 256)          788480
_________________________________________________________________
cu_dnnlstm_6 (CuDNNLSTM)     (None, 200, 128)          197632
_________________________________________________________________
global_max_pooling1d_2 (Glob (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
activation_3 (Activation)    (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 21)                10773
_________________________________________________________________
activation_4 (Activation)    (None, 21)                0
=================================================================
Total params: 35,993,195
Trainable params: 35,993,195
Non-trainable params: 0
_________________________________________________________________
```

- BILSTM



```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_5 (InputLayer)            [(None, 200)]        0

__________________________________________________________________________________________________
embedding_5 (Embedding)         (None, 200, 68)      47400012    input_5[0][0]

__________________________________________________________________________________________________
spatial_dropout1d_5 (SpatialDro (None, 200, 68)      0           embedding_5[0][0]

__________________________________________________________________________________________________
bidirectional_9 (Bidirectional) (None, 200, 1024)    2383872     spatial_dropout1d_5[0][0]

__________________________________________________________________________________________________
bidirectional_10 (Bidirectional (None, 200, 512)     1969152     bidirectional_9[0][0]

__________________________________________________________________________________________________
global_max_pooling1d_9 (GlobalM (None, 1024)         0           bidirectional_9[0][0]

__________________________________________________________________________________________________
global_max_pooling1d_10 (Global (None, 512)          0           bidirectional_10[0][0]

__________________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 1536)         0           global_max_pooling1d_9[0][0]
                                                                 global_max_pooling1d_10[0][0]

__________________________________________________________________________________________________
dense_7 (Dense)                 (None, 512)          786944      concatenate_4[0][0]

__________________________________________________________________________________________________
dropout_5 (Dropout)             (None, 512)          0           dense_7[0][0]

__________________________________________________________________________________________________
dense_8 (Dense)                 (None, 21)           10773       dropout_5[0][0]
==================================================================================================
Total params: 52,550,753
Trainable params: 52,550,753
Non-trainable params: 0
__________________________________________________________________________________________________
```

- BIGRU



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_4 (Embedding)      (None, 200, 50)           34852950

_________________________________________________________________
spatial_dropout1d_4 (Spatial (None, 200, 50)           0

_________________________________________________________________
bidirectional_7 (Bidirection (None, 200, 512)          473088

_________________________________________________________________
bidirectional_8 (Bidirection (None, 200, 256)          493056

_________________________________________________________________
global_max_pooling1d_3 (Glob (None, 256)               0

_________________________________________________________________
dense_5 (Dense)              (None, 512)               131584

_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0

_________________________________________________________________
dense_6 (Dense)              (None, 21)                10773

_________________________________________________________________
activation_3 (Activation)    (None, 21)                0
=================================================================
Total params: 35,961,451
Trainable params: 35,961,451
Non-trainable params: 0
_________________________________________________________________
```

- GRU

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_7 (Embedding)      (None, 200, 50)           34852950
_________________________________________________________________
spatial_dropout1d_7 (Spatial (None, 200, 50)           0
_________________________________________________________________
cu_dnngru_22 (CuDNNGRU)      (None, 200, 256)          236544
_________________________________________________________________
cu_dnngru_23 (CuDNNGRU)      (None, 200, 128)          148224
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 128)               0
_________________________________________________________________
dense_9 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_6 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_10 (Dense)             (None, 21)                10773
_________________________________________________________________
activation_4 (Activation)    (None, 21)                0
=================================================================
Total params: 35,314,539
Trainable params: 35,314,539
Non-trainable params: 0
_________________________________________________________________
```

- LSTM

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_4 (Embedding)      (None, 200, 50)           34852950
_________________________________________________________________
spatial_dropout1d_4 (Spatial (None, 200, 50)           0
_________________________________________________________________
cu_dnnlstm_10 (CuDNNLSTM)    (None, 200, 512)          1155072
_________________________________________________________________
cu_dnnlstm_11 (CuDNNLSTM)    (None, 200, 512)          2101248
_________________________________________________________________
cu_dnnlstm_12 (CuDNNLSTM)    (None, 256)               788480
_________________________________________________________________
dense_10 (Dense)             (None, 128)               32896
_________________________________________________________________
dense_11 (Dense)             (None, 64)                8256
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_12 (Dense)             (None, 21)                1365
=================================================================
Total params: 38,940,267
Trainable params: 38,940,267
Non-trainable params: 0
_________________________________________________________________
```

- CRF-BILSTM

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 200)]        0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 200, 50)      34852950    input_1[0][0]
__________________________________________________________________________________________________
spatial_dropout1d_1 (SpatialDro (None, 200, 50)      0           embedding_1[0][0]
__________________________________________________________________________________________________
bidirectional_1 (Bidirectional) (None, 200, 512)     630784      spatial_dropout1d_1[0][0]
__________________________________________________________________________________________________
bidirectional_2 (Bidirectional) (None, 200, 256)     493056      bidirectional_1[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_1 (GlobalM (None, 512)          0           bidirectional_1[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_2 (GlobalM (None, 256)          0           bidirectional_2[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 768)          0           global_max_pooling1d_1[0][0]
                                                                 global_max_pooling1d_2[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 768)          0           concatenate_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 21)           16149       dropout_1[0][0]
==================================================================================================
Total params: 35,992,939
Trainable params: 35,992,939
Non-trainable params: 0
__________________________________________________________________________________________________
```

- HANLSTM

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_7 (Embedding)      (None, 200, 50)           34852950
_________________________________________________________________
spatial_dropout1d_3 (Spatial (None, 200, 50)           0
_________________________________________________________________
cu_dnnlstm_11 (CuDNNLSTM)    (None, 200, 256)          315392
_________________________________________________________________
cu_dnnlstm_12 (CuDNNLSTM)    (None, 200, 128)          197632
_________________________________________________________________
attention_6 (Attention)      (None, 128)               328
_________________________________________________________________
dense_3 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 21)                10773
=================================================================
Total params: 35,443,123
Trainable params: 35,443,123
Non-trainable params: 0
filters = 512
hidden_dims = 512
embedding_dims = 50
kernel_size = 3
_________________________________________________________________
```

- CGRU

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_2 (Embedding)      (None, 200, 68)           47400012  
_________________________________________________________________
spatial_dropout1d_2 (Spatial (None, 200, 68)           0         
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 200, 1024)         209920    
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 50, 1024)          0         
_________________________________________________________________
cu_dnngru_2 (CuDNNGRU)       (None, 128)               443136    
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0         
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              132096    
_________________________________________________________________
dropout_4 (Dropout)          (None, 1024)              0         
_________________________________________________________________
activation_3 (Activation)    (None, 1024)              0         
_________________________________________________________________
dense_4 (Dense)              (None, 21)                21525     
_________________________________________________________________
activation_4 (Activation)    (None, 21)                0         
=================================================================
Total params: 48,206,689
Trainable params: 48,206,689
Non-trainable params: 0
filters = 512
hidden_dims = 512
embedding_dims = 50
kernel_size = 3

_________________________________________________________________
```

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Aggarwal CC, Zhai CX (2012) Mining text data. Springer Publishing Company, Incorporated
2. Al-Alwani A, Beseiso M (2013) Arabic spam filtering using bayesian model. Int J Comput Appl 79(7):11–14. https://doi.org/10.5120/13752-1582
3. Al-Harbi S, Almuhareb A, Al-Thubaity A, Khorsheed M, Al-Rajeh A (2008) Automatic arabic text classification. In: Proceedings of the 9th international conference on the statistical analysis of textual data, pp 12–14
4. Al Qadi L, El Rifai H, Obaid S, Elnagar A (2019) Arabic text classification of news articles using classical supervised classifiers. In: 2019 2nd International conference on new trends in computing sciences (ICTCS), pp 1–6
5. Al Qadi L, El Rifai H, Obaid S, Elnagar A (2020) A scalable shallow learning approach for tagging arabic news articles. Jordanian J Comput Inform Technol 6(3):263–280
6. Al-Salemi B, Ayob M, Kendall G, Noah SAM (2019) Multi-label arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. Inf Process Manage 56 (1):212–227. https://doi.org/10.1016/j.ipm.2018.09.008
7. Al-Sbou AMF (2018) A survey of arabic text classification models. Int J Elect Comput Eng (IJECE) 8(6):4352–4352. https://doi.org/10.11591/ijece.v8i6.pp4352-4355
8. Al-Shalabi R, Obeidat R (2008) Improving knn arabic text classification with n-grams based document indexing. In: Proceedings of the sixth international conference on informatics and systems, pp 108–112
9. Al-Tahrawi MM, Al-Khatib SN (2015) Arabic text classification using polynomial networks. J King Saud Univ Comput Inform Sci 27(4):437–449. https://doi.org/10.1016/j.jksuci.2015.02.003
10. Al-Zaghoul F, Al-Dhaheri S (2013) Arabic text classification based on features reduction using artificial neural networks. In: Proceedings of the 15th international conference on computer modelling and simulation, pp 485–490
11. Alalyani N, Larabi S (2018) Nada: New arabic dataset for text classification. Int J Adv Comput Sci Appl 9(9):5. https://doi.org/10.14569/ijacsa.2018.090928

12. Alsaleem S (2011) Automated arabic text categorization using svm and nb. Int Arab J eTechnol 2(2):124–128
13. Azarbonyad H, Dehghani M, Marx M, Kamps J (2021) Learning to rank for multi-label text classification: combining different sources of information. Nat Lang Eng 27(1):89–111
14. Bawaneh MJ, Alkoffash MS, Al Rabea AI (2008) Arabic text classification using k-nn and naive bayes. J Comput Sci 4(7):600–605. https://doi.org/10.3844/jcssp.2008.600.605
15. Biniz M, Boukil S, El Adnani F, Cherrat L (2018) Abd elmajid el moutaouakkil, “arabic text classification using deep learning technics. Int J Grid Distrib Comput 11(9):103–114
16. Boudad N, Faizi R (2017) Rachid oulad haj thami, raddouane chiheb, “sentiment analysis in arabic: a review of the literature”. Ain Shams Eng J
17. Dahou A, Xiong S, Zhou J (2016) Mohamed houcine haddoud and pengfei duan, “word embeddings and convolutional neural network for arabic sentiment classication”. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp 2418–2427
18. Dharmadhikari SC, Ingle M, Kulkarni P (2011) Empirical studies on machine learning based text classification algorithms. Adv Comput Int J 2(6):161–169. https://doi.org/10.5121/acij.2011.2615
19. El-Haj M, Rayson P, Aboelezz M (2018) Arabic dialect identication in the context of bivalency and code-switching. LREC 2018, Eleventh International Conference on Language Resources and Evaluation, pp 3622–3627
20. El-Halees A (2007) Arabic text classification using maximum entropy. Islam Univ J (Series of Natural Studies and Engineering) 15:167–167
21. El Kourdi M, Bensaid A, Rachidi Te (2004) Automatic arabic document categorization based on the naïve bayes algorithm. In: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, Semitic ’04, pp 51–58
22. Elnagar A, Einea O (2016) Brad 1.0: Book reviews in arabic dataset. In: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp 1–8. https://doi.org/10.1109/AICCSA.2016.7945800
23. Elnagar A, Khalifa YS, Einea A (2018a) Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications, Springer International Publishing, pp 35–52. https://doi.org/10.1007/978-3-319-67056-0_3
24. Elnagar A, Lulu L, Einea O (2018b) An annotated huge dataset for standard and colloquial arabic reviews for subjective sentiment analysis. Proc Comput Sci 142:182–189. https://doi.org/10.1016/j.procs.2018.10.474 (**arabic Computational Linguistics**)
25. Elnagar A, Einea O, Al-Debsi R (2019) Automatic text tagging of arabic news articles using ensemble learning models”. In: Proceedings of the 3rd International Conference on Natural Language and Speech Processing, pp 59–66
26. Elnagar A, Al-Debsi R, Einea O (2020) Arabic text classification using deep learning models. Inform Process Manag 57 (1):102121–102121. https://doi.org/10.1016/j.ipm.2019.102121
27. Gharib TF, Habib MB, Fayed ZT (2009) Arabic text classification using support vector machines. Int J Comput Their Appl 16 (4):192–199
28. Gonçalves T, Quaresma P, Kłopotek M, Wierzchoń S, Trojanowski K (2004) The impact of nlp techniques in the multilabel text classification problem. Intelligent Information Processing and Web Mining 25
29. Habash N, Hirst G (2010) Introduction to arabic natural language processing. Synthesis Lectures on Human Language Technologies 3(1):1–187. https://doi.org/10.2200/s00277ed1v01y201008hlt010
30. Harrag F, El-Qawasmeh E, Pichappan P (2009) Improving arabic text categorization using decision trees. In: 2009 First International Conference on Networked Digital Technologies, pp 110–115. https://doi.org/10.1109/NDT.2009.5272214
31. Bilal Hawashin A, Mansour Aljawarneh S (2013) An efficient feature selection method for arabic text classification. Int J Comput Appl 83(17):1–6. https://doi.org/10.5120/14666-2588
32. Hmeidi I, Hawashin Bilal, El-Qawasmeh E (2008) Performance of knn and svm classifiers on full word arabic articles. Adv Eng Inform 22(1):106–111. https://doi.org/10.1016/j.aei.2007.12.001
33. Hmeidi I, Al-Ayyoub M, Abdulla NA, Almodawar AA, Abooraig R, Mahyoub NA (2015) Automatic arabic text categorization: A comprehensive comparative study. J Inf Sci 41(1):114–124. https://doi.org/10.1177/0165551514558172
34. Hmeidi I, Al-Ayyoub M, Mahyoub NA, Shehab MA (2016) A lexicon based approach for classifying arabic multi-labeled text. Int J Web Inform Syst
35. Justus D, Brennan J, Bonner S, Mcgough A (2018) Predicting the computational cost of deep learning models, pp 3873–3882. https://doi.org/10.1109/BigData.2018.8622396
36. Korde S, Mahender CN (2012) Text classification and classifiers: a survey. IJAIA J 3(2):85–99. https://doi.org/10.5121/ijaia.2012.3208
37. Li Y, Nie X, Huang R (2018) Web spam classification method based on deep belief networks. Expert Syst Appl 96:261–270. https://doi.org/10.1016/j.eswa.2017.12.016
38. Malmasi S, Dras M (2015) Language identication using classier ensembles. Proceedings of the joint workshop on language technology for closely related languages, varieties and dialects, pp 35–43
39. Mesleh AMA (2007) Chi square feature extraction based svms arabic language text categorization system. J Comput Sci 3 (6):430–435. https://doi.org/10.3844/jcssp.2007.430.435, exported from https://app.dimensions.ai on 2019/02/03
40. Noaman HM, Elmougy S, Ghoneim A, Hamza T (2010) Naive bayes classifier based arabic document categorization. Proceeding of the 7th international conference on informatics and systems (INFOS2010), pp 1–5
41. Raho G, Al-Shalabi R, Kanaan G (2015) Asma’a nassar different classification algorithms based on arabic text classification: Feature selection comparative study. IJACSA) Int J Adv Comput Sci Appl 6(2)
42. Saad M (2010) The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification. Master’s thesis, Computer Engineering Dept., Islamic University of Gaza, Palestine
43. Shehab MA, Badarneh O, Al-Ayyoub M, Jararweh Y (2016) A supervised approach for multi-label classification of arabic news articles. In: 2016 7th International Conference on Computer Science and Information Technology (CSIT), pp 1–6. https://doi.org/10.1109/CSIT.2016.7549465
44. Wang T, Liu L, Liu N, Zhang H, Zhang L, Feng S (2020) A multi-label text classification method via dynamic semantic representation model and deep neural network. Appl Intell 50 (8):2339–2351

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.