**Arabic Sentiment Analysis Using Recurrent Neural Networks: A Review**

#### Abstract

Arabic content generated on websites and in social media has significantly increased over the last decade. In social media, opinions are openly and freely expressed, thereby offering a rich source for trend analysis. These opinions could be mined for valuable indicators so that products or services can be improved. This is accomplished using a natural language processing means such a sentiment analysis. Using deep learning has been increasingly used as a powerful tool for analyzing these opinions owing to its accuracy in predicting unstructured data. Unlike English, Arabic has several specifics that complicate conventional language processing and analysis. Recurrent neural networks (RNNs) is a promising and widely used method in the analysis of sentiment, which exhibits extensive morphological variations. However, we observed that there is no review for RNN Arabic sentiment analysis that presents the feasibility of RNN models for classifying the sentiment. To track RNN models and their effectiveness for Arabic sentiment analysis, we present a systematic review of related state-of-the-art methods and research so that their weaknesses may be identified, thus facilitating the development of future methods. We observed a lack of studies that use RNNs. Therefore, deeper models are required for better representation, as well as Arabic pre-trained embedding models for improved accuracy.

**Keywords** Sentiment analysis · Deep learning · Opinion mining · Sentiment classification · Arabic language · Recurrent neural network

## 1 Introduction

The power and influence of the internet continue to grow globally. According to International Data Corporation, the amount of digital data generated worldwide reached 33 Zettabytes (ZB) in 2018, whereas the expected growth by 2025 is 175 ZB (Reinsel et al. 2018).The number of internet users worldwide has increased to 4.021 billion; in particular, the number of users from the Middle East has increased to 164 million (Kemp 2018). In recent years, an increasing number of people have been using social media to express their opinions, praise a particular service, or vent their frustrations. As of January 2018, the number of social media users is 3.196 billion users, and 130 million of them are Arabs (Kemp 2018). Owing to the influence of social media content on governing, development, diplomacy, and business, sentiment analysis is required in social media monitoring, as it allows an overview of the wider public opinion on topics appearing in a variety of posts, from politics-related to customer reviews. The ability to discern the sentiment and attitude behind a post about any subject facilitates strategizing and future planning, thus providing better services.

Sentiment analysis is the computational assessment of a nation’s attitudes, feelings, and emotions toward structures, persons, cases, news, or subjects. A considerable amount of research has been conducted to improve the precision of sentiment analysis, from basic linear approaches to more complex deep neural network models (Heikal et al. 2018).

Machine learning techniques have been widely used in sentiment analysis. However, these techniques have limited ability to process raw data, and feature representation greatly affects the performance of a machine learner. For this reason, deep learning is used for feature representation at multiple levels. Deep learning automatically discovers discriminative and explanatory text representations from data using nonlinear neural networks, each of which transforms the representation at one level into a representation at a higher and more abstract level (LeCun et al. 2015).

Recently, deep learning has been highly successful in sentiment analysis and is considered a modern multilingual model. However, the accuracy of modern Arabic sentiment analysis still requires improvement. The Arabic language poses several challenges owing to its complex configuration, different dialects, as well as its lack of resources. Even though current deep learning approaches have enhanced the accuracy of Arabic sentiment analysis, it is still possible to improve these approaches (Heikal et al. 2018).

A promising new research field in Arabic sentiment analysis is the application of recurrent neural networks (RNNs) to textual models to demonstrate the learning process and measure the understanding of the text at the level of semantic analysis (Souri et al. 2018). However, a large dataset is required for effectively learning these feature representations.

### Sentiment analysis for Arabic

Sentiment analysis in Arabic is a challenging task, this could be tied to several reasons one of which is the morphology complexity of the language.

### Arabic orthography: Arabic texts is written from right to left and is characterized by the absence upper or lower cases. Its alphabet contains 28 letters: 25 consonants) (ب,ت,ث,ج,ح,خ,د,ذ,ر,ز,س,ش,ص,ض,ط,ظ,ع,غ,ف,ق,ك,ل,م,ن,هـ and only 3 vowels ((أ,و,ي. In addition, short vowels (ُ َ ِ ْ ) are used as diacritical marks. These are put either over or under the letters to indicate the exact articulation and explain the meaning of the text. Moreover, based on the presence or absence of such diacritics, the meaning of words can be totally different. For example, the word علم may mean عِلْم (knowledge), عَلَم (flag), or عَلَّمَ (teach), the letters are identical while diacritics are different, it can be mentioned that most of Arabic words nowdays are written without diacritics (Boudad et al. 2018).

### Arabic morphology: The Arabic language has a highly complicated morphology in which a word may carry significant information. As a space-delimited token, a word in Arabic has numerous morphological features: Inflectional, and Derivational morphology (Saleh 2009; Turki Khemakhem et al. 2010; Boudad et al. 2018). The morphological features can be listed in detail as follows:

* Agglutinative morphology: An Arabic word may be composed of a stem plus one (or more) affixes and clitics. A stem is a combination of a root and derivational morphemes to which an affix (or more) can be added. The Affixes in Arabic are: prefixes, suffixes, and infixes. Prefixes are attached at beginning of the words, where suffixes are attached at the end, and infixes are found in the middle of the words. On the other hand, the clitics include: proclitic, which occur at the most beginning of a word such as (و and) and (ف then), and enclitics that occur at the most ending of a word, which are a complement pronoun. For example, the Arabic word**)** ليكتبونها to write them) contains a number of attached affixes and clitics as shown: (ل: Proclitic), (ي Prefix), (كتب Stem), (ون Suffix), and (ها Enclitic). The complexity nature of Arabic morphology leads to ambiguity in analyzing. For example, the word (وجدنا we found) can be analyzed as (وجد found + نا we) and also as (و and + جد grandfather + نا we).
* Derivational morphology: Derivation is a method for deriving a new word from an existing word, involving a part of speech alteration, a change of meaning, or both; for example, from the root ك ت ب that construct the verb (كَتَبَ wrote), we derive (كَاتِب writer), (مَكْتُوب letter), (كِتَابَة writing), (كِتَاب book) and (مَكْتَب office), that is, one root can generate different words with different meanings which increasing in the ambiguity of analyzing the language.
* Inflectional morphology: It is the variation or change of grammatical form that words undergo to mark distinctions of the case, gender, number, and person. For example, the variation of gender can be noticed in the word (student), it can be written as (طالب) for male and (طالبة) for female. Another word (white) can be written as (أبيض) for male and (بيضاء) for female. The group of these inflected word-shapes is called a lexeme category. To characterize a lexeme, a lemma, which is an exacting shape, is conservatively chosen. The diversity of Arabic words leads them to become a big challenge to NLP.

On the other hand, Arabic has three major forms: Classical Arabic CA, which is the language of the Qur’an (Islam’s Holy Book), Modern Standard Arabic MSA, and dialectical Arabic DA. MSA is the most versatile Arabic language. It is used in formal oral or written communication. Dialectical or colloquial Arabic is used in daily life and exhibits regional variations. In addition,

DA is used in social media, and written texts is written mainly using DA and it has several types based on the region. According to Habash (Habash 2010), there are 30 dialects based on geographical and cultural criteria. Moreover, dialects are generally classified into six basic groups (Habash 2010): Egyptian (includes Egypt and Sudan), Levantine (includes Jordan, Syria, Palestine, and Lebanon), Gulf (includes Saudi Arabia, Oman, Kuwait, United Arab Emirates, Qatar, and Bahrain), North African (includes Morocco, Tunisia, Algeria, Libya, and Mauritania), Iraqi, and Yemenite. For example, the word (كيف حالك how are you) in MSA means in each dialect (Egyptian: إزيك, Levantine: كيفك, Gulf: شلونك, North African: شنوه احوالك, Iraqi: اشونك, Yemenite: كيفوك).

The diversity of dialects along with the richness in scripts, orthography, morphology, phonology, and semantics poses research challenges that require appropriate systems to handle the ambiguity, tokenization, spell checking, stemming, lemmatization, pattern matching, and part-of-speech tagging in the Arabic language.

### Contributions

The contributions of this survey are summarized as follows and in Figure 1.

* + - A systematic review technique is followed to identify recent RNN models for sentiment analysis.
    - The annotated datasets for Arabic languages, which are available online, are discussed for further research.
    - The research studies are explained, categorized by sentiment type (emoji analysis, emotion detection, sentiment classification, and hate speech detection) based on the level (sentence-, document-, and aspect-based).
    - The limitations and drawbacks of the methods are indicated so that alternative approaches may be used.
    - Future research directions are also discussed in the last section.

**Fig. 1**. Survey framework

In Section 2, we present an overview of the systematic literature review method, including the four research questions and the strategies that were used to extract the targeted studies for both English and Arabic. In Sections 3, 4, and 5, we answer the first, second, and third research question, respectively. In Section 6, we describe the evaluation methods for the studies under consideration. Finally, in Section 7, we answer the fourth question and provide a conclusion to our study.

## 2 Overview of the systematic literature review method

The review is divided into two parts: RNNs with English and RNNs with Arabic text. The systematic review that follows the guidelines described by Kitchenham (Kitchenham 2004) and Heckman (Heckman and Williams 2011). The methodology is divided into the following sections: research questions and search strategy for both review parts. The research questions are mentioned in Section 2.1. The search strategy is described in Section 2.2.

### Research questions

In this review, we are interested in the following research questions (RQ). RQ1 will be answered in Section 3 (Neural network approaches for sentiment analysis), RQ2 will be answered in Section 4 (RNN approaches for sentiment analysis), RQ 3 will be answered in Section 5 (Arabic sentiment analysis using RNNs), and RQ4 will be answered in Section 7 (Conclusion and future work). Table 1 indicates these questions and their motivation.

**Table 1.** Research questions of the systematic review with their motivation

|  |  |
| --- | --- |
| Definition | Motivation |
| RQ1: What are the neural network models for sentiment analysis? | To provide background information for systematic techniques used in sentiment analysis. |
| RQ2: What are the types of RNNs for sentiment analysis? | To explore the state-of-the-art techniques and implementations of RNNs in sentiment analysis. We focus on recurrent LSTM and GRUs. |
| RQ3: What are the studies on Arabic sentiment analysis using RNNs? | To review all latest models, techniques, and lexical resources for recurrent models. |
| RQ4: What are the limitations and drawbacks found in the studies? | To investigate the gaps of recurrent models in Arabic. |

### Search strategy

In this section, we present the search terms, search strategy, and databases for the two review parts. In the first review, we consider studies that used RNNs in general. Then, we exclude those that did not use English for sentiment classification and compare them with those that used Arabic. We focus on English, as English is the dominant language for all studies, given the abundance of datasets and available supportive tools. In the second part, we consider studies on Arabic sentiment analysis, as this is the focus of our research, and study potential gaps and challenges. The two parts have different keywords and elimination strategies, as demonstrated in the following sections.

### RNNs for sentiment analysis: After several searches and observations to find the suitable keywords, we used the search sentence “x y”, where x is “recurrent” and “neural”, whereas y is “sentiment” or “opinion.” That is, we focus on all studies that use RNNs for sentiment analysis by combining x and y, resulting in two possible search sentences. We searched in the following databases: Springer, IEEE, ACM, Science Direct, and other databases, such as aclweb, NIPS, AAAI, and Semantic Scholar. The time frame for the search was from 2013 to 2018. The year 2013 was chosen because it was observed that no studies used RNN for sentiment analysis before 2013. Initially, the search resulted in 38,926 papers between 2013 and 2018. The search targeted titles, abstracts, keywords, and full texts to cover all relevant studies. These papers were reviewed for final selection according to the following criteria: Papers related to artificial intelligence were included, whereas irrelevant papers (such as those using non-English for sentiment analysis) were excluded. In addition, those concerned with non-target areas, such as image analysis, video analysis, and gender identification, were excluded as well. After selection, there were 193 papers. Table 2 shows the total number of papers for each database using the mentioned keywords for RNNs in English.

**Table 2.** Number of studies for initial search for each keyword for all databases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Databases Keyword | Springer | IEEE | Science Direct | ACM | | Other |
| Recurrent + neural + sentiment | 691 | 859 | 325 | | 524 | 11,600 |
| Recurrent + neural + opinion | 2,238 | 953 | 1,870 | | 466 | 19,400 |
| Total (initial search) | 2,929 | 1,812 | 2,195 | | 990 | 31,000 |
| Total (after selection) | 37 | 52 | 12 | | 32 | 60 |

### RNNs with Arabic text for sentiment analysis: In this part of the search, the focus was on studies concerned with RNNs for Arabic sentiment analysis, and we used the combination of three search terms. The first two terms were the same x and y as in the previous section, and the additional term “Arabic” is combined with the two search sentences. We searched in the following databases: Springer, IEEE, ACM, Science Direct, and other databases such as aclweb, NIPS, AAAI, and Semantic Scholar. Initially, the keyword search resulted in 2,636 papers between 2013 and 2018. The search targeted titles, abstracts, keywords, and full texts to cover all relevant studies. The papers were reviewed for final selection according to the criteria used in the previous section. In addition, papers on non-Arabic sentiment analysis were excluded. After selection, there were 24 papers. By comparing these studies with those in the previous section, we conclude that using RNN for sentiment classification for Arabic text is still in its infancy. Figure 2 shows the number of studies on Arabic sentiment analysis using RNN for each year. Table 3 shows the total number of papers for each database using the mentioned keywords for Arabic text.

**Fig. 2**. Number of studies that use RNNs for Arabic sentiment analysis

**Table 3.** Number of studies for initial search for each keyword for all databases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Databases Keyword | Springer | IEEE | Science Direct | ACM | Other |
| Arabic+ recurrent + neural + sentiment | 38 | 65 | 37 | 26 | 807 |
| Arabic + recurrent + neural + opinion | 95 | 53 | 55 | 20 | 1,440 |
| Total (initial search) | 133 | 118 | 92 | 46 | 2,247 |
| Total (after selection) | 4 | 5 | 5 | 0 | 10 |

A review on related study (Yadav and Vishwakarma 2019) indicates that RNNs are one of the best and most important methods used in for text-based sentiment analysis while CNN shown good results for image sentiment. RNNs have frequently achieved high accuracy in both English and Arabic languages compared to shallow neural models. RNN models take a text dataset as their input and attempt to correctly classify each sentence through training. In the output, RNN methods generate novel text sequences according to their knowledge process. The success of the learning process increases as the training data increase. In spite of the importance and effectiveness of RNNs in sentiment analysis in various languages, including Arabic, this field has not been studied in depth. Therefore, in this study, RNNs have been chosen as a method that can be accurately and effectively used for sentiment analysis in the Arabic language.

## 3 Neural network approaches for sentiment analysis

In this section, we will answer RQ1. To understand recent trends in sentiment analysis using neural networks, we explain in the next three subsections the principles of sentiment analysis and its importance, the techniques that have previously been used to obtain a sentiment analysis system, and the classification levels. In Subsection 3.4, we explore word embedding (a well-known technique in sentiment analysis) to extract features and represent text to be classified. In Subsection 3.5, we discuss neural networks and deep learning. Subsequently, we review related techniques that have been used for sentiment analysis, including deep learning, in Subsection 3.6.

### Sentiment analysis

Sentiment analysis or opinion mining is a computational process for studying opinions, sentiments, and attitudes about products, services, or topics that are expressed using text. This is carried out by identifying and categorizing these opinions and extracting subjective information using natural language processing and machine learning techniques (Medhat et al. 2014). Starting from the 2000s, a large amount of unstructured data has been generated daily, and sentiment analysis has become an essential research field using natural language processing as well as text and data mining; it is widely applied in areas, such as politics, commerce, tourism, education, and health, to evaluate services and improve productivity mining reviews, survey responses, and social media (Zhang et al. 2018). According to Hemmatian and Sohrabi (Hemmatian and Sohrabi 2017) and Rani and Kumar (Rani and Kumar 2019), sentiment analysis involves collecting reliable data from various resources, identifying the opinion for each phrase, preparing labeled data for analysis, and extracting features by classifying these phrases.

### Techniques

There are two techniques for sentiment analysis systems (Cortes and Vapnik 1995; Madhoushi et al. 2015): machine learning and lexicon-based; furthermore, there are hybrid methods that use both machine learning and lexicon-based techniques.

### Machine learning techniques: These techniques can be supervised, unsupervised, or semi-supervised. They are classified into “shallow learning” approaches, such as naive Bayes (NB), maximum entropy (ME), and support vector machines (SVMs), and “deep learning” approaches using neural networks for feature learning and sentiment classification. SVMs (Cortes and Vapnik 1995) are machine learning models that are used for classification and regression. An SVM is a large-margin, non-probabilistic, linear classifier. The principle of SVMs is to determine a hyperplane that separates training data points into two classes and keeps the margin, which is the distance between the hyperplane and the nearest data point, as large as possible, considering the correct side of each data point. Thus, an SVM makes decisions based on the nearest data points, which are called support vectors and are selected as the only effective elements in the training set. Pang et al. (Pang et al. 2002) were the first to use machine learning for sentiment classification. In that study, SVMs, NB, and ME were used to classify movie reviews into two classes (positive and negative) using unigrams. It was demonstrated that machine-learning models perform better than simple counting methods but could not perform as well in sentiment classification as in traditional topic-based categorization, with an accuracy of roughly 83%.

### Lexicon-based techniques: These techniques can be divided into “dictionary-based” and “corpus-based” approaches. The system does not require training to classify the data. Instead, it has predetermined sentiment values to compare the features in the text (Vohra and Teraiya 2013). A sentiment lexicon dictionary contains lists of words that express feelings and opinions; it uses scores for the opinion words to count the most dominated lexicons and determine whether a sentence is positive or negative (Liu and Zhang 2012). By contrast, a corpus relies on syntactic or co-occurrence patterns and has a large collection of texts for other opinion words. The technique starts with a list of seed opinion adjectives and a set of linguistic constraints such as *and, but, either-or, neither-nor* to explore additional adjective opinion words and their orientations (Liu and Zhang 2012).

### Hybrid techniques: Hybrid approach combines both lexicon-based and machine learning-based approaches such that the trained model considers the lexicon based result in its features as in (Alhumoud et al. 2015), they examined machine learning algorithms SVM and KNN in relation to the performance of the hybrid approach. Their experiment confirmed that the use of the hybrid approach yields better accuracy.

### Classification levels

Sentiment analysis classifications can be applied at document, sentence, and aspect granularity level (Zhang et al. 2018). At document level, the entire document is classified as either positive or negative. In sentence-level sentiment classification, each sentence in the document is classified as positive, negative, or neutral. At aspect level, product features are identified and extracted from the source data, where entity and aspect/feature extraction is performed along with aspect sentiment classification.

### Word embedding

In sentiment analysis, words could be numerically represented before classification. Word embedding is a technique for language modeling and feature learning to enhance sentiment analysis (Zhang et al. 2018) in which features are contextually learnt, and words are converted into real-valued vectors with lower dimensionality. The power of this technique is that words with similar meaning are represented as similar vectors. There are several available techniques for word embedding; some of the most common are briefly represented in the following. The first is using manual feature extraction, such as Bag of Words (BoW) (Zhang et al. 2010). This is a challenging and time-consuming task, particularly in the Arabic language, which is morphologically rich. Moreover, the BoW method cannot capture the relationships between words or words that have the same meanings, and bigram and trigram approaches are required to handle this. The second is using Word to Vector (word2vec) (Mikolov et al. 2013), which contains the continuous BoW (CBOW), and the Skip–Gram (SG) models. The CBOW model predicts the target word from the surrounding words within a window’s length. In contrast, The SG is model used to predict the surrounding words from the target word. The third is using Global Vector (GloVe) (Pennington et al. 2014), which is trained on the nonzero entries of a global word-word co-occurrence matrix. Thus, unlike word2vec, which is predictive model, GloVe counts the word frequency in a context.

Most existing word embedding techniques (including those mentioned here) model only the syntactic context and ignore sentiment. However, there are some Arabic word embedding techniques that are employed for sentiment classification (Altowayan and Tao 2016; Soliman et al. 2017; Alayba et al. 2018a).

AraVec (Soliman et al. 2017) is an example of Arabic word embedding for sentiment classification. Six different word embedding models were constructed for the Arabic language using three different resources[[1]](#footnote-1): Wikipedia, Twitter, and Common Crawl webpage crawl data. Two models for each resource and the SG model were provided. These models were evaluated using qualitative and quantitate measures on several tasks that involved capturing word similarity.

Alayba et al. (Alayba et al. 2018a) presented an Arabic word embedding model using a 1.5-billion-word corpus. Different Word2Vec models were constructed using the Abu El-Khair Corpus (El-khair 2016) to choose the most suitable for the study. CBOW with 200 dimensions was chosen for an automatic Arabic lexicon. It was used with different machine learning methods and convolutional neural networks and was compared with different feature selection methods sentiment classification. In addition, the health service dataset3 was used to test the generated model. Compared with the previous study (Alayba et al. 2017) on the health service dataset, this approach increased the sentiment classification from 0.85 to 0.92 for the Main dataset, and from 0.87 to 0.95 for the Sub-dataset.

Altowayan and Tao (Altowayan and Tao 2016) used the CBOW model for word representation learning using a large Arabic corpus[[2]](#footnote-2) that includes completed texts of the Quran, standard Arabic from news articles, the Arabic edition of international networks, and dialectal Arabic from consumer reviews, with a total of 159,175 vocabulary items. To test the model, twitter and book reviews datasets for sentiment classification, and news articles datasets for subjectivity classification were used with six different machine learning classification models. A comparison of the subjectivity classification model with handcrafted models (Banea et al. 2010; Mourad and Darwish 2013) demonstrated that the proposed model outperforms the handcrafted models on the same dataset.

Fasttext[[3]](#footnote-3) is another word embedding and text classification method, particularly in the case of rare words. It uses character level information and supports 294 languages.It uses sub word information in the prediction model.

### Artificial neural network and deep learning

Artificial neural networks are complex networks with several neurons distributed in layers that simulate the function of human brain (Schmidhuber 2015). A neural network consists of an input layer and an output layer and may include hidden layers of non-linear processing units that link neurons. Moreover, a neural network learns features depending on the real-valued activations and suitable weights that make the neuron exhibit desired behaviors (Schmidhuber 2015).

Neural networks may be feedforward or recurrent/recursive (Arabiyat 2017). Feedforward neural networks use a straightforward data processing scheme from the input layer through a hidden layer to the output layer. In the hidden layers, there are no cycles, and thus the output of any layer does not affect that same layer. These networks use a backpropagation algorithm (BP) to train and compute a gradient of the cost function using the recent input to update the network parameters and thereby reduce errors during training. In contrast, recurrent/recursive neural networks contain a loop. Therefore, they can process data from prior connections/values, as well as input from the recent layer to predict the output of the current layer. It uses BP through time, which is a regular backpropagation but calculates the gradient of a cost function using all inputs, not only the recent inputs.

Deep learning is a machine learning technique that uses a neural network with multiple deep layers for data processing, and thus it learns complex from simpler features as it proceeds from lower to higher layers using real-number activations for each neuron and weights for each link (Schmidhuber 2015; Arabiyat 2017).

### Sentiment analysis using deep learning

In recent years, deep learning has achieved satisfactory results in natural language processing, speech recognition, and computer vision (Young et al. 2018). In sentiment analysis, several types of models using deep neural networks have been employed (Ain et al. 2017), such as convolutional neural networks (CNNs) (Kim 2014), RNNs (Kobayashi et al. 2010) including deep bidirectional recurrent neural networks (Deep Bi-RNN) (Irsoy and Cardie 2014), long short-term memory (LSTM) (Tai et al. 2015), gated recurrent unit (GRUs) (Tang et al. 2015), recursive autoencoders (RAEs) (Socher et al. 2011), recursive neural networks (Socher et al. 2013), and hybrid methods. In the present study, we will focus on recurrent LSTM and GRUs. Figure 3 shows the classification techniques for sentiment analysis.

SVM

ME

Dictionary

Corpus

Sentiment analysis

Lexicon-based

Machine-learning-based

Deep learning

NB

CNN

Recurrent

Feedforward

Unidirectional

Bidirectional

Bi-GRU

Bi-LSTM

LSTM

GRU

Hybrid

Shallow learning

Hierarchal

Recursive NN

Hybrid

**Fig. 3.** Sentiment analysis techniques

One of the challenges in sentiment analysis using machine learning methods is the lack of correctly labeled data owing to difficulties related to subjective interpretation and expensive human labor. These difficulties hinder the size of trained data, affect performance, and can lower classification accuracy (Pang and Lee 2008). To address these issues, sentiment analysis using deep learning techniques has merged owing to their automatic learning capability. This allows algorithms to understand sentence structure and semantics and to generate new feature representations in contrast to traditional methods that choose the most frequent word in the given input.

Given the importance of sentiment analysis using deep learning, numerous related studies involving English texts have been conducted. However, few studies have been published focusing on Arabic sentiment analysis.

## Recurrent neural network approaches for sentiment analysis

This section answers RQ2. To consider the latest trends in RNNs for sentiment analysis we first explain the concept of an RNN. We will present an RNN as well as its architecture and function; moreover, we will introduce bi-directional RNNs and other models, such as LSTM and GRUs, which are crucial to understand related studies that will be reviewed in the next section.

### RNNs

RNNs (Elman 1990; Kobayashi et al. 2010; Ayyadevara 2018) take several input vectors and output other vectors, but in the case of sentiment analysis, RNNs output only one vector, which is either positive or negative (Ayyadevara 2018). RNNs share learned features and perform the same task for every element of a sequence; they use previous computations to compute the current output. The inputs are a hidden state vector for the previous timestamp (*t*-1) and the input state vector at a certain time (*t*). Equation (1) generates the hidden state vector (*h*), and *W*, *U*, and *b* are parameter metrics, where (*Wh*) is a weight matrix used to condition the input (*Xt*). (*Uh*) is a weight matrix used to condition the previous hidden state (*ht*-1). The result of the activation function in Equation (1) is passed to next timestamp and to *softmax* activation to generate the output in Equation (2). Figure 4 shows the architecture of an RNN for three timestamps, and Table 4 defines the symbols appearing in Equations (1) and (2).

|  |
| --- |
| (1) |
| (2) |

RNNs are used for named entity recognition in a given text. There are several problems with basic RNNs (Bengio et al. 1994). Specifically, they use an earlier sequence to make a prediction, and this can lower accuracy. This problem can be resolved using Bi-RNNs (Schuster and Paliwal 1997). Moreover, the exploding gradient problem may occur during the training process when unstable weights and directions are generated. This can be resolved using gradient clipping (Pascanu et al. 2013). Finally, the vanishing gradient problem, in which long-term dependencies cannot be captured, may also occur. This can be resolved using LSTM and GRUs.

**Table 4.** Symbols and their definitions for RNN equations

|  |  |
| --- | --- |
| Symbol | Definition |
| *ht* | Hidden state vector for current time step. |
| *ht*­1 | Hidden state vector for previous time step. |
| *Xt* | Input state vector for current time step. |
| *yt* | Output for current time step. |
| *Wh*, *Uh*, *b* | Parameter metrics, where *Wh* and *Uh* refer to the weights assigned to the hidden state vectors, and *b* refers to the bias. They are randomly initialized and are learnt as the network trains. |
| *tanh* | Hyperbolic tangent function. The output range is [-1, 1]. |
| *softmax* | Function that is used in the final layer of the network and transforms numbers into probabilities that sum to one. |

**Fig. 4.** RNN architecture

### Bi-RNNs

Schuster and Paliwal first proposed Bi- RNNs (Schuster and Paliwal 1997). Their principle is that the output at each timestamp depends on the preceding as well as the succeeding elements in the sequence. A Bi-RNN consists of two RNNs proceeding in opposite time directions. This approach can increase the amount of input information available to the network, as information from both past and future states is used. Hence, more accurate outputs are obtained at each timestamp. Figure 5 shows the architecture of a Bi-RNN (Wang et al. 2015). The symbol *x* is the input state vector, *h* is the hidden state vector, and *y* is the output.

**Fig. 5.** Bi-RNN architecture

### LSTM

LSTM (Hochreiter and Schmidhuber 1997; Graves et al. 2013) is an RNN that uses a complex activation function to capture long-term dependencies confidently and resolve the vanishing gradient problem using three gates. Along with the input state, it has a hidden and a cell state. In addition to calculating the activation function and output probability at each timestamp, LSTM calculates the forget, input, and output gates using the sigmoid function 𝜎, whose values range from 0 to 1. In Equation (3), the forget gate 𝑓 is used to determine what information be discarded from the cell state *C*. If the result is approximately zero, the information for the cell state will be discarded; otherwise, it will be retained.

|  |
| --- |
| *ft= 𝜎 ( Wf ht­1+Uf Xt + bf )* (3) |

In the next step, it is determined what new information will be added to the cell state using the input gate *I*, as seen in Equation (4). If there is no update, the value is zero.

|  |
| --- |
| *it = 𝜎 ( Wi ht­1+Ui Xt + bi )* (4) |

In Equation (5), a *tanh* layer generates a vector of new candidate values *C̃*. Subsequently, in Equation (6), the cell state *C* is updated using Equations (3), (4), and (5).

|  |
| --- |
| C̃*t = tanh ( Wc ht­1+Uc Xt + bc )* (5) |
| *Ct = Ct­1\* ft + it \**  (6) |

Finally, the new hidden state vector *h* is generated. In Equation (7), the output gate *O* is used to determine which parts of the cell state will be output using the sigmoid function. Subsequently, using the *tanh* function of the new cell state *C*, the hidden state is obtained in Equation (8). Figure 6 illustrates the LSTM architecture (Olah). Table 5 defines the symbols appearing in Equations (3), (4), (5), (6), (7), and (8) for more clarity.

|  |
| --- |
| *Ot =𝜎 ( Wo ht­1+Uo Xt + bo )* (7) |
| *ht = Ot \* tanh ( Ct )* (8) |

**Table 5.** Symbols and their definitions for LSTM equations

|  |  |
| --- | --- |
| Symbol | Definition |
| *ht* | Hidden state vector for current time step. |
| *ht*­1 | Hidden state vector for previous time step. |
| *Xt* | Input state vector for current time step. |
| *ft* | Forget gate for current time step. |
| *it* | Input gate for current time step. |
| *Ot* | Output gate for current time step. |
| C̃*t* | Candidate value for current time step. |
| *C* | Cell state vector for current time step. |
| *Ct*­1 | Cell state vector for previous time step. |
| *Wf*, *Wi*,*Wc* ,*Wo*,*Uf*, *Ui*,*Uc* ,*Uo*,*bf*,*bi* , *bc* ,*bo*. | Parameter metrics, where *W* and *U* refer to the weights (e.g. *Wo* is a weight assigned to the output gate *O*). *b* refers to the bias (e.g. *bo* is the bias assigned to the output gate). They are randomly initialized with random numbers and are learnt as the network trains. |
| *𝜎* | Sigmoid function, which takes a real-valued number and returns a value in the range [0, 1]. |
| *tanh* | Hyperbolic tangent function. This takes a real-valued number, and the output range is [-1, 1]. |

**Fig. 6.** LSTM architecture

### GRU

A GRU is a variation on LSTM. Choe et al. (Cho et al. 2014) proposed an RNN encoder–decoder and introduced the GRU model, which uses two gates (instead of three gates in LSTM) and fewer parameters, and thus it is a simpler model. The two gates are the reset gate *rt*, which indicates the relevance of the previous cell state for computing the next candidate, as shown in Equation (11), and the update gate *ut*, which is a combination of the forget and the input gates, as seen in Equation (10). Initially, the cell state equals the hidden state, that is, a *tanh* layer generates a vector of new candidate values *C̃* in Equation (9) using the reset gate *rt*. Subsequently, in Equation (12), the cell state *C* is updated using Equations (9) and (10). The GRU architecture is shown in Figure 5 (Hochreiter and Schmidhuber 1997).

|  |
| --- |
| (9) |
| (10) |
| (11) |
| (12) |

As *Ct* = *Ct*­1 and the update gate *ut* is always zero, GRU does not suffer from vanishing gradients problem, thus allowing the RNN to learn long dependencies. Table 6 defines the symbols appearing in Equations (9), (10), (11), and (12). Figure 7 shows the GRU architecture.

**Table 6.** Symbols and their definitions for GRU equations

|  |  |
| --- | --- |
| Symbol | Definition |
| *Ct* | Cell state vector for current time step. |
| *Ct*­1 | Cell state vector for previous time step. |
| *Xt* | Input state vector for current time step. |
| *rt* | Reset gate for current time step. |
| *ut* | Update gate for current time step. |
| C̃*t* | Candidate value for current time step. |
| *Wu*, *Wr*,*Wc* ,*Uu*, *Ur*,*Uc* ,*bu*,*br* , *bc*. | Parameter metrics, where *W* and *U* refer to the weights (e.g., *Uu* is the weight assigned to the update gate *O* to condition the current input state), and *b* refers to the bias (e.g., *bu* is the bias assigned to the update gate). They are randomly initialized and are learnt as the network trains. |
| *𝜎* | Sigmoid function, which takes a real-valued number and returns a value in the range [0, 1]. |
| *tanh* | Hyperbolic tangent function, which takes a real-valued number, and its output range is [-1, 1]. |

**Fig. 7.** GRU architecture

According to Chung et al. (Chung et al. 2014), there are similarities and differences between LSTM and GRUs. The most prominent common feature is their ability to retain existing components and add the updated component instead of replacing the entire activation function as in conventional recurrent units. Accordingly, long-term dependencies may be captured, and shortcut paths may be created so that the error can be easily backpropagated without rapidly vanishing, thus eliminating the vanishing gradient problem.

A difference between LSTM and GRUs is that the former controls the exposure of memory content and cell state to other units using the output gate, whereas the latter expose the entire cell state to other units without control. Another difference, as mentioned earlier, is that the LSTM unit has separate input and forget gates, whereas the GRU merge these operations through the reset gate. The LSTM unit does not control the information flow from the previous timestamp, but it controls the amount of the new content being added to the memory cell from the forget gate. By contrast, a GRU controls the information flow from the previous activation when computing the new candidate activation, but does not control the amount of the candidate activation being added, as the control is performed through the update gate.

## Arabic sentiment analysis using RNNs

This section answers RQ3 by discussing related studies that use RNN models for Arabic sentiment analysis. There are only 24 such studies. We include those studies that introduced a new Arabic dataset for sentiment analysis and used RNN to test the accuracy of these corpora. As Arabic datasets are scarce and limited, we highlight the various datasets that were used in all studies under consideration in Table 7.

We divide these papers according to the classification level (sentence, aspect-based) and the type of analysis (emotion detection, emoji analysis, hate speech detection, and sentiment classification) that were applied.

**Table 7.** Related studies that used Arabic datasets for sentiment analysis using RNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datasets | Size | Classes | Arabic form | Papers |
| LABR (Aly and Atiya 2013). | 63,257 book reviews. | 3-class | MSA/Dialectal. | (Abbes et al. 2017).  (Baniata and Park 2016). |
| SemEval-2016 Task 5 (Pontiki et al.). | ~3000 hotel reviews. | 4-class | MSA/Dialectal. | (Tamchyna and Veselovská 2016).  (Ruder et al. 2016a).  (Wang and Lu 2018).  (Ponti et al. 2017).  (Al-Smadi et al. 2018).  (Al-Smadi et al. 2019). |
| ASTD (Nabil et al. 2015). | 10,006 tweets. | 4-class | Egyptian dialect. | (Al-Azani and El-Alfy 2017).  (Al-Azani and El-Alfy 2018).  (Baccouche et al. 2018).  (Alayba et al. 2018b).  (Heikal et al. 2018). |
| ArTwitter (Abdulla et al. 2013). | 2000 tweets. | 2-class | MSA/Jordanian dialect. | (Al-Azani and El-Alfy 2017).  (Alayba et al. 2018b).  (Al-Azani and El-Alfy 2018). |
| QCRI (Mourad and Darwish 2013). | 2300 tweets. | (objective, subjective), 2-class | MSA/Egyptian, Levantine, and Gulf dialects. | (Al-Azani and El-Alfy 2018). |
| Syria (Mohammad et al. 2016). | 2000 tweets. | 2-class, 3-class | Syrian dialect. | (Al-Azani and El-Alfy 2018). |
| Semeval-2017 Task 4 (Rosenthal et al. 2017). | 1656 (2-class) + 3355 tweets. | 2-class, 3-class 5-class | Dialectal. | (Al-Azani and El-Alfy 2018).  (González et al. 2017).  (Samy et al. 2018). |
| SemEval-2018 Task 1 (Mohammad et al. 2018). | 11781 tweets. | 7-class,12-emotion classes | Dialectal. | (Abdullah and Shaikh 2018).  (Abdullah et al. 2018).  (Samy et al. 2018).  (Alhuzali et al. 2018).  (Abdou et al. 2018). |
| DINA (Abdul-Mageed et al. 2016). | 3000 tweets. | 6-emotion classes | Dialectal. | (Alhuzali et al. 2018). |
| AraSenTi (Al-Twairesh et al. 2017). | 17,573 tweets. | 2-class | Saudi dialect. | (Alwehaibi and Roy 2018). |
| MD-ArSenTD | 14400 tweets. | 3-class | Dialectal. | (Baly et al. 2017b). |
| BRAD 2.0 | 692586 book reviews. | 3-class | MSA/Dialectal. | (Elnagar et al. 2018). |
| Arabic Health Services Dataset | 2026 tweets. | 2-class | Dialectal. | (Alayba et al. 2018b). |
| Collected tweets. | 12,897 tweets.  6600 tweets. | 2-class  2-hate class | Dialectal  Dialectal. | (El-Kilany et al. 2018).  (Albadi et al. 2018). |
| ArSenL | 157,969 synsets | 3-class | MSA | (Badaro et al. 2014). |
| Semeval-2016 Task 7 | 1366 tweets | 4-class | MSA/Dialectal | (Kiritchenko et al. 2016). |
| Arabic Gold Standard Twitter Data for Sentiment Analysis | 4,191 tweets | 3-class | MSA/Dialectal | (Refaee and Rieser 2014) |
| ArSAS | 19,762 tweets | 3-class | MSA/Dialectal | (Elmadany et al. 2018) |

### Aspect-based level sentiment analysis using RNN

In this section, we focus on studies that classify the text on an aspect level. These works are categorized based on the type of RNN that was used.

Al-Smadi et al. (Al-Smadi et al. 2018) proposed an aspect-based sentiment analysis of Arabic hotel reviews using two implementations: deep RNN and SVM, along with word embedding, lexical, syntactic, morphological, and semantic features. The F1 score was employed to evaluate aspect opinion target expression (OTE) extraction (T1) and aspect category identification (T2), and the accuracy score to evaluate sentiment polarity identification (T3). Different models were used for performance comparison: a baseline model (AL-Smadi et al. 2016), INSIGHT-1 (Ruder et al. 2016b), and UFAL (Tamchyna and Veselovská 2016). The results demonstrated that SVM outperformed the RNN approach in all tasks due to the rich hand-crafted feature set extracted to train the SVM model, particularly in T3, with an accuracy of 95.4%, compared with RNN with an accuracy of 87%. However, deep RNN outperformed SVM in terms of execution time, particularly in T2, with a speed rate of 5.3× compared with T1, with s speed rate of 4.4×, and T3, with speed rate of 2.1×.

Tamchyna and Veselovska (Tamchyna and Veselovská 2016) employed a multilingual LSTM model for aspect-based sentiment analysis (ABSA) using the categories consumer electronics, restaurants, and hotel reviews in seven languages: Arabic, Dutch, English, French, Russian, Spanish, and Turkish. The main goal was to discover the linguistic patterns in the data automatically without feature extraction or language-specific tools. The model was compared with five different versions of the same model: Official baseline (provided by the task organizer) using SVM, baseline model using logistic regression without any type of feature except BOW, a model submitted for official evaluation, where the networks were not fully optimized by the submission deadline, an optimized model, where the results were obtained after the deadline, and the best model, where the best score for each language and domain was reported. Regarding the Arabic language, the optimized model exhibited the best results in terms of the F1 score (52.59%), compared with all the other models, particularly the best model, with an F1 score of 52.11%.

Al-Smadi et al. (Al-Smadi et al. 2019) improved the deep learning approach in (Al-Smadi et al. 2018) for aspect-based sentiment analysis of Arabic hotel reviews using LSTM: First, a character-level bidirectional LSTM along with a conditional random field classifier (Bi-LSTM-CRF) for aspect-OTE extraction. Secondly, an aspect-based LSTM for aspect sentiment polarity classification in which the aspect-OTEs are considered attention expressions to support sentiment polarity identification. For sentiment polarity identification, the authors compared the proposed model with the baseline model (AL-Smadi et al. 2016), INSIGHT-1 (Ruder et al. 2016b) and IIT-TUDA (Kumar et al. 2016). The authors used the F1 score to evaluate OTE extraction, and the accuracy to evaluate sentiment polarity identification. The results demonstrated that this approach outperformed the baseline on both tasks, with an improvement of 39% for aspect-OTE extraction, and 6% for aspect sentiment polarity classification. For OTE extraction, the proposed model yielded promising results, particularly when FastText was used for character embedding, with an F1 score of 69.98%, compared with the same model when word2vec was used, which had an F1 score of 66.32%. Moreover, for sentiment polarity identification, the proposed model outperformed other models, with an accuracy of 82.6%, except INSIGHT-1, which uses CNN and had an accuracy of 82.7%.

Ruder et al. (Ruder et al. 2016a) presented a hierarchical bidirectional LSTM that is able to consider intra-sentence relations such as background and inter-sentence relations (e.g., ABSA using bidirectional sentence-level LSTM and bidirectional review-level LSTM). The evaluation was performed using seven different models: the best model for each domain and language (Pontiki et al. 2016) XRCE (Brun et al. 2016), IIT-TUDA (Kumar et al. 2016), a sentence-level CNN (INSIGHT-1) (Ruder et al. 2016b), a sentence-level LSTM, which is the first layer of the proposed model, and the proposed model with randomly initialized word embeddings (H-LSTM) and with pre-trained embeddings (HP-LSTM). Regarding the Arabic language, the proposed model with pre-trained word embedding outperformed other models, including sentence-level models, with an accuracy of 82.9% compared with the best model, which had an accuracy of 82.7%.

Wang and Lu (Wang and Lu 2018) proposed a segmentation attention-based Bi-LSTM model for ABSA that extracts sentiment information and learns latent opinions by capturing the structural dependencies between the given target and the sentiment expressions using a CRF layer. The proposed model was tested on two groups of datasets: One was from online reviews (Laptop, Restaurant) and social comments on Twitter and SemEval 2014 Task 4 was used to analyze each component of the model. In the other group, the SemEval 2016 Task 5 dataset (Pontiki et al.) was used to examine the model’s language sensitivity. The evaluation was performed by comparing the extracted opinions with the manually annotated opinions using several models from Ruder et al. (Ruder et al. 2016a), the LSTM with standard attention *softmax* (A-LSTM), and the proposed model with segmentation attention layer with and without penalty terms (SA-LSTM and SA-LSTM-P). Regarding the Arabic hotel reviews, the experiments demonstrated that the proposed model with penalty achieved the best results, with an accuracy of 86.9%, compared with other models, particularly with the proposed model without penalty, which had an accuracy of 86.7%.

Ponti et al. (Ponti et al. 2017) examined the amount of information that representations retain about the polarity of sentences for each language. Moreover, they presented a model that decodes sentiment from unsupervised sentence representations learnt by different architectures (either sensitive to word or sensitive order, or not), such as additive Skip-Gram, FastSent, sequential denoising AutoEncoder (SDAE), and distributed BOW (DBOW) models, and they compared these models with bi-directional LSTM. Regarding the Arabic language, DBOW yielded the best results compared with other unsupervised representations, with a weighted F1 score of 76.76%, whereas SDAE had an F1 score of 72.13%. However, Bi-LSTM outperformed all unsupervised strategies, with an F1 score of 86.56%, note that Bi-LSTM used SkipGram model for sentence representation, hence, an RNN models can be chosen as a ceiling especially for Arabic language.

Table 8 shows the results for aspect-based sentiment analysis for Arabic text using RNNs.

**Table 8.** Aspect-based sentiment analysis for Arabic text using RNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papers | Classifier | Word embedding | Dataset | Accuracy |
| (Al-Smadi et al. 2018). | RNN. | Word2vec. | SemEval-2016 Task 5 : Arabic Hotel reviews (Pontiki et al.). | 87%. |
| (Tamchyna and Veselovská 2016). | LSTM. | Word2vec: CBOW. | SemEval-2016 Task 5 including Arabic Hotel reviews (Pontiki et al.). | F1 score: 52.59%. |
| (Al-Smadi et al. 2019). | LSTM. | Word2vec and FastText for character level embedding. | SemEval-2016 Task 5 : Arabic Hotel reviews (Pontiki et al.). | LSTM: 82.6% |
| (Ruder et al. 2016a). | Hierarchical Bi-LSTMs. | Trained embedding by Leipzig Corpora Collection. | SemEval-2016 Task 5 including Arabic Hotel reviews (Pontiki et al.). | LSTM: 80.5%.  H-LSTM: 82.8%.  HP-LSTM: 82.9%. |
| (Wang and Lu 2018). | Bi-LSTM + CRF | Trained embedding by Leipzig Corpora Collection. | SemEval-2016 Task 5 including Arabic Hotel reviews (Pontiki et al.). | A-LSTM: 86.5%.  SA-LSTM: 86.7%.  SA-LSTM-P: 86.9%. |
| (Ponti et al. 2017). | Bi-LSTM. | Additive SkipGram, ParagraphVec DBOW, FastSent, Sequential Denoising AutoEncoder SDAE. | SemEval-2016 Task 5 including Arabic Hotel reviews (Pontiki et al.). | Weighted F1 score:  86.56%. |

It should be noticed that no study used GRU, Bi-GRU, or hybrid methods for aspect-level classification. In addition, multilingual models as shown in (Tamchyna and Veselovská 2016) weren’t keen to improve Arabic language specifically. Moreover, all studies depend on a unified dataset, demonstrating the need for well-annotated datasets for aspect terms, categories, and sentiment polarity.

### Sentence-level sentiment analysis using RNN

In this part, we will divide the studies according to the type of analysis that is involved, namely, emotion detection, emoji analysis, hate speech detection, and sentiment classification.

### *5.2.1 Emotion detection*

This part will introduce studies that are concerned with the detection and classification of emotions such as anger, fear, joy, and sadness.

Samy et al. (Samy et al. 2018) proposed GRU and context-aware GRU (C-GRU) to investigate the role of social influence on shaping others' opinions and emotions in the same environment and the effect on the determination of sentiment polarity. These models extracted the contextual information (topics) and used both topic and sentence information to detect multi-label emotions. C-GRU was compared with a support vector classifier (SVC) with a linear kernel, L1 regularization from (Badaro et al. 2018), and context-free GRU. The results demonstrated that C-GRU outperformed the other model with an accuracy of 53.2%, F1-macro average of 64.8%, and F1-micro average of 49.5% compared with simple GRU, which had an accuracy of 52.4%, F1-macro average of 64.2%, and F1-micro average of 49.6%.

Abdullah and Shaikh (Abdullah and Shaikh 2018) presented the Team UNCC’s system, which used an LSTM network to detect emotion intensity or sentiment in English, Arabic, and translated Arabic tweets using SemEval 2018. The system attempted to complete all five subtasks and determine the intensity and sentiment of tweets in SemEval 2018. The tweets were preprocessed and fed into word2vec, doc2vec, and other feature vectors for feature extraction. Subsequently, these vectors were fed into the deep neural network layers for prediction. The Spearman correlation scores demonstrated that the proposed system yielded promising results, with an emotion regression of 59.7%, emotion classification of 51.7%, sentiment regression of 77.3%, and sentiment classification of 74.8%, compared with the baseline model by the SemEval-Task 1’s organizers, which is based on SVM-Unigrams and had an emotion regression of 45.5%, emotion classification of 31.5%, sentiment regression of 57.1%, and sentiment classification of 47.1%

Abdullah et al. (Abdullah et al. 2018) presented a new version of TeamUNCC’s system, called SEDAT, to explore the emotion intensity and sentiment for Arabic tweets. This model consisted of two sub-models: the first used a collection of Arabic tweets with five dimensions and translated tweets with 4903 dimensions using a set of features to produce vectors, and the other used only the Arabic tweets with 300 dimensions, which are trained using skip gram-twitter from AraVec to produce vectors. The first sub-model passed generated vectors to a fully connected neural network, whereas in the second sub-model, the vectors were fed into CNN-LSTM. The output layer consisted of one sigmoid neuron that produced a real-valued number between 0 and 1. The Spearman correlation scores demonstrated that SEDAT outperformed TeamUNCC’s system (Abdullah and Shaikh 2018) with an emotion regression of 66.1%, emotion classification of 56.9%, sentiment regression of 81.7%, and sentiment classification of 78.6%; moreover, SEDAT is only 1 to 2 points behind the state-of the art models, that is, AffecThor for emotion (58.7%) and EiTAKA for sentiment (80.9%) (Mohammad et al. 2018).

Abdou et al. (Abdou et al. 2018) presented AffecThor, which consists of three different models. These models used learned and manually crafted representations, such as character embedding, word embedding, inference, and average lexicon representations. The proposed models used a feed-forward DNN or gradient boosted trees for regression, and two ensemble regressors: simple averaging cross-validation and a nonlinearity (sigmoid) layer on top of the different sub-models, such as CNN and Bi-LSTM. The use of simple averaging provides the best results for all SemEval 2018 models in Arabic language, particularly in emotion classification, with an emotion classification of 58.7%.

Alhuzali et al. (Alhuzali et al. 2018) described and physically confirmed a technique for the mechanical procreation of emotionally labeled information; an improved dataset was also used to extract modern and dialectical Arabic emotions, which focused on eight types of Robert Plutchik’s core emotions. Using a mixed supervision method that exploits the seeds of first-person feelings, it was also demonstrated that promising results can be obtained through an RNN with deep gates. Alhuzali et al. extended a manually annotated dataset from Dina (Abdul-Mageed et al. 2016), called LAMA. The proposed dataset was based on emotion existence and intensity in one stage for English and Arabic. The proposed approach is based on emotion phrase seeds from Robert Plutchik’s eight basic emotion types: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. In addition, the authors proposed a hybrid supervised method that automatically determines emotion intensity and sentiments for English and Arabic, and they compared it with supervised and distant supervised methods. The proposed method was applied with baseline models, such as multinomial NB (MNB), passive aggressive classifier (PAC), perceptron classifier (PTN) SVM, SVM trained with stochastic gradient descent (SVM-SGD), and the proposed GRU. In addition, the models were validated using SemEval-2018 (Mohammad et al. 2018). The results demonstrated that GRU yielded the best results, particularly in terms of emotion detection, on all datasets, particularly the Dina and Lama-Dina datasets, with an F1 score of 98%, compared with SVM-SGD, which had an F1 score of 92%. Moreover, a hybrid supervision of the datasets Lama-Dina+Lama-Dist using GRU yielded the best results in terms of emotion classification, with an average F1 score of 70% compared with SVM-SGD, which had an F1 score of 60%. Table 9 shows the emotion detection models for Arabic text.

**Table 9.** Sentence/document/word level sentiment analysis based on emotion detection for Arabic text

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Papers | Classifier | Sentiment Classification Level | Document/Text Representation | Dataset | Accuracy |
| (Samy et al. 2018). | GRU, C-GRU. | Sentence level. | AraVec(Soliman et al. 2017). | From SemEval-2017 (Rosenthal et al. 2017) and SemEval-2018 (Mohammad et al. 2018). | GRU: 53.2%.  C-GRU: 52.4%. |
| (Abdullah and Shaikh 2018). | Dense and LSTM networks. | Sentence and document level. | Word2vec, doc2vec with different feature vectors. | From SemEval-2018 (Task 1: effect in Tweets) (Mohammad et al. 2018): raw Arabic and Arabic translated into English. | Emotion detection: 44.6%.  Spearman correlation scores: Emotion classification 51.7 %.  Sentiment classification 74.8%. |
| (Abdullah et al. 2018). | CNN-LSTM | Sentence and document level. | AraVec(Soliman et al. 2017), doc2vec and a set of semantic features. | From SemEval-2018 (Task 1: effect in Tweets) (Mohammad et al. 2018): raw Arabic and Arabic translated into English. | Spearman correlation scores: Emotion classification 56.9%.  Sentiment classification 78.6%. |
| (Abdou et al. 2018). | CNN, Bi-LSTM | Sentence and word level. | Word2vec skip-gram embeddings. | From SemEval-2018 (Task 1: effect in Tweets) (Mohammad et al. 2018). | Spearman correlation scores: Emotion classification 58.7%.  Sentiment classification 75.2%. |
| (Alhuzali et al. 2018). | GRU. | Sentence level. | Word2vec. | LAMA, DINA (Abdul-Mageed et al. 2016), DIST and SemEval-2018 (Mohammad et al. 2018). | LAMA-DINA+LAMA-DIST: F1 score  70%. |

As noticed from Table 9, there is a lack of research on emotion detection and analysis using neural nets. Moreover, it is noted that there is a dominant dataset that is used for emotion analysis studies, namely, SemEval-2018 (Task 1: affect in Tweets).

### *5.2.2 Emoji analysis*

This part introduces studies on emoji analysis using RNN. Emoji are ideograms and smileys that are used frequently in the social media to express ideas and emotions. They differ from emoticons, which use letters, numbers and symbols to create icons.

Al-Azani and El-Alfy (Al-Azani and El-Alfy 2018) used deep RNN models, namely, LSTM, Bi-LSTM, GRU, and Bi-GRU, with different modes (summation, multiplication, concatenation, and average of outputs) for emoji-based tweets to detect sentiment polarity, and compared these models with deep neural networks and baseline machine learning classifiers, such as stochastic gradient descent, Gaussian NB, SVM, k-nearest neighbors, and decision trees. These models used a set of 843 Arabic microblogs with emoji from different resources, such as Arabic Sentiment Tweets Dataset (ASTD) (Nabil et al. 2015), ArTwitter (Abdulla et al. 2013), QCRI (Mourad and Darwish 2013), Syria (Mohammad et al. 2016), and Semeval-2017 Task4 Subtask#A (Rosenthal et al. 2017); in addition, data were collected from Twitter and YouTube and were manually annotated. They used the Emoji Sentiment Ranking (ESR) lexicon to recognize emojis in the dataset. The performance was evaluated in terms of precision, recall, F1 measure, accuracy, MCC correlation coefficient, and geometric mean (GM). The results demonstrated that LSTM and GRU models significantly outperformed other models. Specifically, the bidirectional GRU performed best, with an accuracy of 78.71% and an F1 score of 78.76%, compared with Bi-LSTM, which had an accuracy of 77.99% and an F1 score of 78.10%.

Baccouche et al. (Baccouche et al. 2018) proposed an automatic labeling technique for health-related tweets in three different languages: English, French, and Arabic. In addition, they applied sentiment analysis models such as CNN and LSTM to classify tweets into two and three classes. The dataset contained both a domain-specific health-related dataset and a non-specific domain dataset from Amazon, IMDB, Yelp, and ASTD. The automatic labeling technique preprocessed the health-related tweets to detect the annotation using domain knowledge, NLP, and sentiment-lexicon dictionaries. The proposed annotation models yielded the best results using RNN, particularly for English, with an accuracy of 98% and an F1 score of 97%, compared with CNN, which had an accuracy of 97% and an F1 score of 96%. Moreover, the proposed model performed better by 1.1% when a non-specific domain dataset was added to LSTM.

Heikal et al. (Heikal et al. 2018) presented a model that combined CNN and Bi-LSTM to predict the sentiment of Arabic tweets using the ASTD dataset. This model followed the same method as in a previous study for English (Cliche 2017), using various hyper-parameters to improve accuracy and handling emoticons by mapping them to the related emojis. Although the proposed model did not use any feature engineering to extract special features, it achieved the best results, with an accuracy of 65.05% and an F1 score of 64.46% compared with the state-of-the-art model from Baly et al. (Baly et al. 2017a), which had an accuracy of 58.5% and an F1 score of 53.6%. Table 10 summarizes the studies that focus on emoji analysis for Arabic text.

**Table 10.** Emoji analysis for Arabic text

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papers | Classifier | Document/Text Representation | Dataset | Accuracy |
| (Al-Azani and El-Alfy 2018). | LSTM, Bi-LSTM, GRU, Bi-GRU. | Emoji Sentiment Ranking (ESR) lexicon. | Combined datasets from ASTD (Nabil et al. 2015), ArTwitter(Abdulla et al. 2013), QCRI (Mourad and Darwish 2013), Syria (Mohammad et al. 2016), Semeval-2017 Task4 Subtask#A(Rosenthal et al. 2017) and other resources | Bi-GRU: 78.71%.  Bi-LSTM: 77.99%. |
| (Baccouche et al. 2018). | LSTM. | Word2vec. (pretrained by Wikipedia). | Health related dataset (authors didn’t mention the final number), and non-health-related dataset from Amazon, IMDB, Yelp and ASTD dataset (Nabil et al. 2015). | 83%. |
| (Heikal et al. 2018). | CNN-Bi-LSTM. | AraVec(Soliman et al. 2017). | ASTD dataset (Nabil et al. 2015). | CNN-LSTM:  65.05%. |

Only three studies performed emoji analysis with the best result of 83%, highlighting the need for more research in this direction. As emoji are used in a paradoxical way, a robust model that detects sarcasm and actual sentiment is required. In addition, there is a scarcity of datasets containing emojis. Accordingly, data are collected manually.

### *5.2.3 Hate speech detection*

Only one study (Albadi et al. 2018) focused on hate speech detection in Arabic. This is a new topic, and the study introduced the first dataset for addressing this issue.

Albadi et al. (Albadi et al. 2018) provided the first dataset for detecting religious hate speech in Arabic tweets. It consists of 6,000 labeled tweets (Albadi). Moreover, they created the first three Arabic lexicons consisting of common terms in religious discussions, with scores that describe the polarity and strength of these terms. In addition, they developed three different approaches to capture religious hate speech: lexicon-based using three Arabic lexicons, n-gram-based using logistic regression and SVM, and deep-learning-based using GRU with AraVec embedding(Soliman et al. 2017). The GRU model achieved the best results compared with all other models, with an accuracy of 79%, an F1 score of 77%, and an area under the receiver operating characteristic curve (AUROC) of 84%. The second-best model (SVM) had an accuracy of 75%, F1 score of 72%, and AUROC of 81%.

### *5.2.4 Sentiment classification*

In this section, we divide the related studies into two parts. First, studies that used LSTM, GRU, or both. Second, hybrid models, where an additional neural network was used as a layer in the model along with an RNN such as CNN. They are presented in the following.

*LSTM/GRU models*. Abbes et al. (Abbes et al. 2017) proposed two models based on deep learning sentiment analysis: a deep neural network (DNN) and an RNN for Arabic social media. The proposed models followed a number of steps. The first stage was gathering and collecting data from the Large-scale Arabic Book Reviews Dataset (LABR) (Aly and Atiya 2013). The second stage was the preprocessing and building of a lexicon model. The third phase was the feature extraction step, consisting in deriving the lexicon-based relevant features from the stored data using word embedding. The final steps involved applying deep learning models, that is, DNN and RNN (LSTM), and classifying sentence polarity. The experimental results demonstrated that RNN outperformed DNN, with an accuracy of 71%, precision of 68.3%, recall of 77%, and F1 score of 72.4%, whereas DNN had an accuracy of 64.4%, precision of 61.1%, recall of 75.3%, and F1 score of 67.5%. Although the unfair comparison done using 300 epoch for DNN and only 30 epoch for RNN, the highest result was at epoch 200 for DNN and epoch 12 for DNN.

Alwehaibi and Roy (Alwehaibi and Roy 2018) presented LSTM to classify Arabic texts with different pre-trained word embedding techniques, namely, AraVec, ArabicNews, and Arabic FastTxt (AraFT), to investigate the effect of these techniques on the accuracy of the model. Initially, the model preprocessed the AraSenTi tweet datasets. Subsequently, the tweets were processed by a pre-trained word embedding technique to generate vectors for each word. Then, the embedding was fed to the LSTM layer with a 128-dimensional hidden state to classify each tweet into positive, negative, or neutral. AraFT achieved the best accuracy (93.5 %) compared with AraVec (88%) and ArabicNews (91%). Although the dataset is divided equally for three classes, the low F1 score (43% for ArabicNews, 40% for AraVec, and 41% for AraFT) is an indication of both poor precision and poor recall.

Baly et al. (Baly et al. 2017b) provided the ﬁrst multi-dialect Arabic sentiment Twitter dataset (MD-ArSenTD). It contains annotated tweets (for both sentiment and dialect) that were collected from 12 Arab countries from the Gulf, Levant, and North Africa. In addition, the authors analyzed tweets from Egypt and the United Arab Emirates (UAE) to investigate the regional variation of data characteristics and their effect on sentiment by using feature engineering for SVM and generic and dialect-speciﬁc embeddings for LSTM. The results demonstrated the superiority of LSTM over SVM, particularly when lemma embedding was used for Egyptian tweets, with an accuracy of 70.0% and weighted F1 score of 69.1%, compared with UAE tweets, with an accuracy of 63.7% and weighted F1 of 64.8% for 3-class classification.

Elnager et al. (Elnagar et al. 2018) presented book reviews in the Arabic dataset BRAD 2.0, which is an extension of BRAD 1.0 with more than 200,000 additional records and a total of 692586 annotated reviews, and it combines both MSA and dialect Arabic with three classes: positive, negative and neutral. In the experiment, only two classes have been used. The authors applied NB, decision tree, random forest, XGBoost, SVM, CNN and LSTM to verify and validate the proposed dataset. Machine learning classifiers generally performed better using unigram and bigram, especially without TF-IDF, while SVM achieved the best result with TF-IDF with an accuracy of 90.86%. Using deep learning models, the results show that LSTM outperforms CNN, especially when using GloVe, with an accuracy of 90.05% compared with CNN with an accuracy of 90.02%. However, imbalanced classified dataset leads to biased for positive class; with 64% of data have been classified as positive, 15% as negative, and 21% as neutral.

El-Kilany et al. (El-Kilany et al. 2018) presented a model that recognized the sentiment targets from Arabic tweets using two layers: a character and word2vec embedding layer, and a bidirectional LSTM with a CRF classification layer. The Bi-LSTM used vectors produced by the first layer to generate the features for CRF, which used these contextual features to predict a tag for the current word based on the previous and consequent word tags in the sentence. The results demonstrated that the proposed model achieved higher recall (71.4%) compared with the same model that used only character-embedding without word2vec, which achieved 61.5%. This implies that it discovered more target entities from the tweets and conserved almost the same extraction precision (approximately 73%).

Table 12 indicates the sentence/character-level sentiment analysis using LSTM/GRU for Arabic text.

**Table 12.** Sentence/character-level sentiment analysis using LSTM/GRU for Arabic text

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Papers | Classifier | Classification Level | Document/Text Representation | Dataset | Accuracy |
| (Abbes et al. 2017). | LSTM. | Sentence level. | TF-IDF, BOW. | LABR (Aly and Atiya 2013). | 64.4%. |
| (Alwehaibi and Roy 2018). | LSTM. | Sentence level. | AraVec ArabicNews, and Arabic FastTxt (AraFT). | AraSenTi dataset (Al-Twairesh et al. 2017). | AraFT: 93.5 %.  AraVec: 88%.  ArabicNews: 91%. |
|  |  |  |  |  |  |
| (Baly et al. 2017b). | LSTM. | Sentence level. | Word2vec, lemma embedding, and stem embedding: skip-gram. | MD-ArSenTD. | Egyptian tweets: 70.0%  UAE tweets: 63.7%. |
| (Elnagar et al. 2018). | LSTM. | Sentence level. | GloVe(Pennington et al. 2014), TF-IDF. | BRAD 2.0 (Elnagar). | 90.05%. |
| (El-Kilany et al. 2018). | Bi-LSTM + CRF. | Sentence and character level. | Word2vec and character embedding. | 3000 tweets (available upon request). | F1 score: 72.6%. |

*Hybrid models.* Al-Azani and El-Alfy(Al-Azani and El-Alfy 2017) compared various CNN and LSTM approaches for sentiment analysis of Arabic microblogs using six models: LSTM, CNN, CNN with LSTM, three-stacked LSTM layers, and two LSTMs combined with summation, multiplication, and concatenation. These models were evaluated for Arabic sentiment analysis using four evaluation measures: precision, recall, accuracy, and F1. Two benchmark Arabic tweet datasets were used: ASTD (Nabil et al. 2015) and Arabic sentiment analysis ArTwitter (Abdulla et al. 2013). Word2vec was used as input to the investigated models, with static and non-static word initialization for CBOW and skip-gram word embedding. The experiments demonstrated that using word2vec vectors updated during learning achieved the best results in nearly all cases. In addition, LSTM outperformed CNN. Moreover, non-static models with the combined LSTM architectures performed better than other models, particularly when two LSTMs were combined with concatenation along with ArTwitter dataset and with skip-gram word embedding. The results demonstrated that precision reached 87.36%, recall 87.27%, accuracy 87.27%, and F1 score 87.28% compared with the same architecture for the ArTwitter dataset and with CBOW word embedding, which had a precision of 86.46%, recall of 86.45%, accuracy of 86.45%, and F1 score of 86.45%.

Alayba et al. (Alayba et al. 2018b) presented a model that combined a CNN and LSTM for sentiment classification at character, character 5-gram, and word levels to expand the number of features, and they investigated its accuracy using different datasets such as main and sub Arabic Health Services Dataset (Alayba et al. 2017), ArTwitter (Abdulla et al. 2013), and ASTD (Nabil et al. 2015). Moreover, the proposed model was compared with those by Alayba et al. (Alayba et al. 2018a), Abdulla et al. (Abdulla et al. 2013), and Dahou et al. (Dahou et al. 2016). Although the model by Dahou et al. achieved the best results for the ASTD dataset, with an accuracy of 79.07%, the proposed model achieved a promising accuracy, particularly with at the character 5-gram level and with the sub Arabic health dataset, with an accuracy of 95.68%, compared with the word-level model on the main Arabic health dataset, with an accuracy of 94.24%. The high accuracies are due to the unbalanced datasets for main and sub Arabic Health Services Dataset. Although the character, character 5-gram, and word level reached a results of 74.19%, 77.62%, and 76.41% respectively for ASTD dataset, the proposed model from (Al-Azani and El-Alfy 2017) showed a highest results with LSTM-MUL with CBOW with an accuracy of 81.63%, showing the importance of word embedding techniques over the row embedding techniques.

Gonzalez et al. (González et al. 2017) proposed the ELiRF-UPV system for enhancing Task 4 of SemEval-2017. It involved five different subtasks: (A) message polarity classification, (B) and (C) tweet classification according to a two and five-point scale, respectively, and (D) and (E) tweet quantification according to a two and five-point scale, respectively. The proposed model used a CNN and an RNN and the combination of general and specific word embeddings with polarity lexicons using both English and Arabic. The system used three CRNNs (including a CNN, max pooling, and Bi-LSTM) to extract spatial relations among the words of a sentence with an input of three different embeddings: out-domain embeddings, in-domain embeddings, and sequences of word polarities. The outputs of these networks were used as an input to a fully connected multilayer perceptron. Different measurements made to evaluate each task. Regarding subtask A, the proposed model using Arabic achieved an accuracy of 50.8% compared with English, where an accuracy of 59.9% was obtained.

Baniata and Park (Baniata and Park 2016) proposed a combined CNN and Bi-LSTM in two different models using Arabic. The first model consisted of three convolutional layers with filter sizes 3, 4, and 5 followed by the average pooling Bi-LSTM layer. The third layer was a merged layer that was connected to a fully connected layer including a multi perception layer and a sigmoid classifier. The second model was an inversed model starting with the Bi-LSTM layer, which was connected with three convolutional layers with filter sizes 3, 4 and 5 followed by a fully connected layer, which was a sigmoid classifier. The results demonstrated that the CNN-Bi-LSTM achieved better sentence feature representation, with an accuracy of 86.43% compared with Bi-LSTM-CNN, which had an accuracy of 66.26%.

Table 13 shows the sentence/character-level sentiment analysis using hybrid neural networks for Arabic text.

**Table 13.** Sentence/character level sentiment analysis using hybrid networks for Arabic text

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Papers | Classifier | Sentiment Classification Level | Document/Text Representation | Dataset | Results |
| (Al-Azani and El-Alfy 2017). | LSTM, CNN-LSTM, Stacked-LSTM, Combined-LSTM-SUM, Combined-LSTM-MUL, Combined-LSTM-CONC. | Sentence level. | Word2vec. | Two datasets of Arabic tweets: ASTD (Nabil et al. 2015) and ArTwitter(Abdulla et al. 2013). | LSTM-CONC with ArTwitter and Skip-gram:  87.27%.  LSTM-MUL with ASTD and CBOW: 81.63%. |
| (Alayba et al. 2018b). | CNN-LSTM. | Character level, character 5-gram level, and word level. | Row embedding based on the level. | Arabic Health Services Dataset (Alayba et al. 2017), ArTwitter(Abdulla et al. 2013), and ASTD (Nabil et al. 2015). | Main-AHS: 94.24%.  Sub-AHS: 95.68%  ArTwitter: 88.10%  ASTD: 77.62%. |
| (González et al. 2017). | CNN-Bi-LSTM. | Sentence level. | Word2vec. | SemEval-2017 Task 4 (Rosenthal et al. 2017). | (task A):  Arabic: 50.8%. |
| (Baniata and Park 2016). | CNN-Bi-LSTM, Bi-LSTM-CNN. | Sentence level. | polyglot (Al-Rfou et al. 2013). | LABR (Aly and Atiya 2013). | CNN-Bi-LSTM: 86.43%. |

As noticed, no research has been conducted on Arabic sentiment analysis using GRU, Bi-LSTM, or Bi-GRU as a stacked model for comparison with stacked models using LSTM. Although there is a lack of character-level analysis, there are promising results. Moreover, Arabic pre-trained word embedding strongly affects the analysis, which requires a well- trained embedding for word and character level. Moreover, using word embedding concurrently with character embedding could enhance the accuracy as shown in (El-Kilany et al. 2018).

Table 14 presents the reviewed algorithms, their advantages, and their drawbacks in addition to assessments.

**Table 14.** Types of algorithms that have been reviewed in related works

|  |  |  |  |
| --- | --- | --- | --- |
| Model type | Advantages | Drawbacks | Assessment |
| RNN | It can utilize the same transport function with the same parameters in each phase | Difficulty in training data, high gradients at several phases, and not suitable for image or tabular datasets. | It is a type of ANN, and is particularly helpful in translations and sentiment analysis. |
| SVM | It is most efficient in high-dimensional areas or when the number of dimensions is better than the number of samples, and it is relatively memory-efficient. | It is not suitable for large data sets and is not effective when the dataset contains more noise, so that the target groups overlap. | It is one of the most accurate algorithms in sentiment analysis, but it cannot be used to analyze large databases. |
| LSTM | One of the most successful RNN algorithms, it overcomes repeated network training problems and is capable of learning long-term dependencies. | It cannot be stacked in deep models, and it cannot track long-term dependencies. | The repeating module for LSTM is more complicated. Instead of a single neural network layer, there are four layers interacting in a special manner. |
| CRF | It is perfect for various segmentation and sequence tagging tasks such as sentiment analysis | It is computationally complex in the training phase, and retraining is difficult when new training data samples are available. | It is a supervised machine learning algorithm; therefore, it requires a sufficiently large training sample. |
| GRU | GRU is superior to LSTM, as it can be trained in less time and more effectively. Moreover, it is simple to modify and does not require memory modules. | It cannot extract local context features. | It is an enhanced LSTM algorithm that has been improved in terms of network architecture and effectiveness, but it does overcome the inherent defect of LSTM in capturing local text features. |

## Evaluation Metrics

The following standard evaluation metrics have been used to measure the efficiency of the previous approaches (Ruder et al. 2016c; Tamchyna and Veselovská 2016; Baniata and Park 2016; Abbes et al. 2017; Al-Azani and El-Alfy 2017, 2018; Baly et al. 2017a; González et al. 2017; Ponti et al. 2017; Abdullah and Shaikh 2018; Alayba et al. 2018b; Alhuzali et al. 2018; El-Kilany et al. 2018; Elnagar et al. 2018; Heikal et al. 2018; Samy et al. 2018; Wang and Lu 2018; Abdou et al. 2018; Al-Smadi et al. 2018, 2019; Albadi et al. 2018; Abdullah et al. 2018; Alwehaibi and Roy 2018; Baccouche et al. 2018).

### Accuracy

It is used to evaluate the accuracy of a model, and it is defined as the percentage of correctly classified tweets to the total tweets. Therefore, it can be represented mathematically as follows:

### Precision

It evaluates the strictness of the classifier output. Precision is the percentage of tweets classified as positive correctly to the total number of samples classified as positive. It can be calculated as follows:

### Recall

We use recall to measure the integrity of a classifier’s output. It measures the percentage of actual tweets that were correctly classified and can be calculated as follows:

### F1 score

It combines precision and recall as follows:

In (13) until (16),

TP (true positive) is the number of tweets that were correctly classified as positive,

TN (true negative) is the number of tweets that were correctly classified as negative,

FP (false positive) is the number of tweets that were incorrectly classified as positive, and

FN (false negative) is the number of tweets that were incorrectly classified as negative.

## Research gaps

This part is to answer Q4 by presenting the findings from the related studies. Sentiment analysis is a difficult ​​research area with diverse and complicated tasks. The most studied tasks are subjectivity categorization, sentiment categorization, lexicon making, feature mining, feature sentiment classification, and attitude spam recognition.

Using RNNs for sentiment analysis has yielded accurate results, as these networks use previous sequential states to compute the current input, which is suitable for natural language context (Elman 1990; Bengio et al. 1994; Kobayashi et al. 2010; Ayyadevara 2018; Souri et al. 2018).

We observed the lack of studies that use RNN for Arabic sentiment analysis compared to other languages, such as English. As of 2018, the number of studies in English reaches 193 while in Arabic, only 24 studies in this field. Moreover, in (González et al. 2017; Abdullah and Shaikh 2018; Abdullah et al. 2018), sentiments were analyzed in both Arabic and English. However, in (Tamchyna and Veselovská 2016), sentiments were analyzed using seven languages: Arabic, Dutch, English, French, Russian, Spanish, and Turkish, whereas (Baccouche et al. 2018) analyzed the data in three different languages: English, French, and Arabic. This demonstrates that using Arabic in this task is promising and still in its infancy, particularly at the document and character level, where there are only six studies on Arabic aspect level analysis and five studies on emotion analysis. (Samy et al. 2018) used GRU and C-GRU, but in (Alhuzali et al. 2018), a hybrid model was developed using supervised algorithms for automatically determining the emotion intensity and feelings for English and Arabic. GRU was demonstrated to yield the best results, particularly in emotion detection and compared with SVM-SGD. It was also found that the use of supervised algorithms with GRU yields an average F1 score of 70% on emotion rating compared with 60% by SVM-SGD. There was only one study on hate speech detection (Albadi et al. 2018) and no studies on sarcasm or cyberbullying detection. This demonstrates the need of further research in that direction. Moreover, constructing deeper models may enhance classification accuracy.

In addition, developing a pre-trained word and character embedding enhances the accuracy of sentiment classification, covering the different types of feature engineering such as Part of Speech POS, NER, and handling negation for different dialects. Based on the study by Alwehaibi and Roy (Alwehaibi and Roy 2018), it was found that character embedding shown promising results especially when combining it with word embedding as shown in (Al-Smadi et al. 2019), (Abdou et al. 2018), and (El-Kilany et al. 2018).

We observed the large number of Arabic datasets with small size, this may occurred due to the diversity of Arabic dialect, nevertheless, create a unified dataset for each dialect may help for fair comparison. Moreover, create a well annotated dataset for different classification area is required for aspect-based analysis purpose.

## Conclusions and future work

In this paper, we presented a systematic review of RNNs for Arabic sentiment analysis in addition to answering four research questions. These questions covered the neural network approaches for sentiment analysis, the RNN approaches for sentiment analysis, and the related studies that use RNN for Arabic sentiment analysis along with the research gaps.

The future direction is to develop Arabic word embedding models that deals with negation words, and without ignoring word structure, to focus on models that utilize emoji for analysis, and to enhance models for document and aspect classification. Several algorithms can be combined for efficient sentiment analysis in large databases. An improved model needs to be developed to analyze Arabic sentences by using a wide range of unsupervised methods. A hybrid model needs to be constructed in which Arabic terminology can be analyzed considering grammatical composition and semantic accuracy. Various CNN and RNN algorithms need to be developed and trained with a variable number of parallel torsion layers. Furthermore, the problem of developing CNNs needs to be modeled as a metaheuristic improvement task of constructing a system of classification of Arabic sentiment. Finally, models need to be optimized so that they may be applied to various applications, such as image classification, object detection, and big data classification.

## References

Abbes M, Kechaou Z, Alimi AM (2017) Enhanced Deep Learning Models for Sentiment Analysis in Arab Social Media. In: Liu D, Xie S, Li Y, et al. (eds) Neural Information Processing. Springer International Publishing, Cham, pp 667–676

Abdou M, Kulmizev A, Ginés i Ametllé J (2018) AffecThor at SemEval-2018 Task 1: A cross-linguistic approach to sentiment intensity quantification in tweets. In: Proceedings of The 12th International Workshop on Semantic Evaluation. Association for Computational Linguistics, New Orleans, Louisiana, pp 210–217

Abdulla NA, Ahmed NA, Shehab MA, Al-Ayyoub M (2013) Arabic sentiment analysis: Lexicon-based and corpus-based. In: 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). IEEE, Amman, Jordan, pp 1–6

Abdullah M, Hadzikadicy M, Shaikhz S (2018) SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp 835–840

Abdullah M, Shaikh S (2018) TeamUNCC at SemEval-2018 Task 1: Emotion Detection in English and Arabic Tweets using Deep Learning. In: Proceedings of The 12th International Workshop on Semantic Evaluation. Association for Computational Linguistics, New Orleans, Louisiana, pp 350–357

Abdul-Mageed M, Alhuzali H, Abu Elhija D, Diab M (2016) Dina: A multidialect dataset for arabic emotion analysis. In: The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media. p 29

Ain QT, Ali M, Riaz A, et al (2017) Sentiment Analysis Using Deep Learning Techniques: A Review. thesai, pp 424–433

Alayba AM, Palade V, England M, Iqbal R (2018a) Improving Sentiment Analysis in Arabic Using Word Representation. In: 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR). IEEE, London, pp 13–18

Alayba AM, Palade V, England M, Iqbal R (2017) Arabic Language Sentiment Analysis on Health Services. 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR) 114–118. https://doi.org/10.1109/ASAR.2017.8067771

Alayba AM, Palade V, England M, Iqbal R (2018b) A Combined CNN and LSTM Model for Arabic Sentiment Analysis. In: Holzinger A, Kieseberg P, Tjoa AM, Weippl E (eds) Machine Learning and Knowledge Extraction. Springer International Publishing, Cham, pp 179–191

Al-Azani S, El-Alfy E-S (2018) Emojis-Based Sentiment Classification of Arabic Microblogs Using Deep Recurrent Neural Networks. In: 2018 International Conference on Computing Sciences and Engineering (ICCSE). IEEE, Kuwait City, pp 1–6

Al-Azani S, El-Alfy E-SM (2017) Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs. In: Liu D, Xie S, Li Y, et al. (eds) Neural Information Processing. Springer International Publishing, Cham, pp 491–500

Albadi N Religious Hate Speech Detection for Arabic Tweets. In: https://github.com/nuhaalbadi/Arabic\_hatespeech. https://github.com/nuhaalbadi/Arabic\_hatespeech

Albadi N, Kurdi M, Mishra S (2018) Are they Our Brothers? Analysis and Detection of Religious Hate Speech in the Arabic Twittersphere. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, Barcelona, pp 69–76

Alhumoud S, Albuhairi T, Altuwaijri M (2015) Arabic sentiment analysis using WEKA a hybrid learning approach. In: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K). pp 402–408

Alhuzali H, Abdul-Mageed M, Ungar LH (2018) Enabling Deep Learning of Emotion With First-Person Seed Expressions. In: PEOPLES@NAACL-HTL

Al-Rfou R, Perozzi B, Skiena S (2013) Polyglot: Distributed Word Representations for Multilingual NLP. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning. Association for Computational Linguistics, pp 183–192

Al-Smadi M, Qawasmeh O, Al-Ayyoub M, et al (2018) Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels’ reviews. Journal of Computational Science 27:386–393. https://doi.org/10.1016/j.jocs.2017.11.006

AL-Smadi M, Qwasmeh O, Talafha B, et al (2016) An enhanced framework for aspect-based sentiment analysis of Hotels’ reviews: Arabic reviews case study. In: 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST). IEEE, Barcelona, Spain, pp 98–103

Al-Smadi M, Talafha B, Al-Ayyoub M, Jararweh Y (2019) Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. International Journal of Machine Learning and Cybernetics 10:2163–2175. https://doi.org/10.1007/s13042-018-0799-4

Altowayan AA, Tao L (2016) Word embeddings for Arabic sentiment analysis. In: 2016 IEEE International Conference on Big Data (Big Data). IEEE, Washington DC,USA, pp 3820–3825

Al-Twairesh N, Al-Khalifa H, Al-Salman A, Al-Ohali Y (2017) AraSenTi-Tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets. Procedia Computer Science 117:63–72. https://doi.org/10.1016/j.procs.2017.10.094

Alwehaibi A, Roy K (2018) Comparison of Pre-Trained Word Vectors for Arabic Text Classification Using Deep Learning Approach. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp 1471–1474

Aly M, Atiya A (2013) LABR: A Large Scale Arabic Book Reviews Dataset. Unpublished. https://doi.org/10.13140/2.1.3960.5761

Arabiyat A (2017) Automatic Arabic Text Diacritization Using Recurrent Neural Networks. The University of Jordan

Ayyadevara VK (2018) Recurrent Neural Network. In: Ayyadevara VK (ed) Pro Machine Learning Algorithms : A Hands-On Approach to Implementing Algorithms in Python and R. Apress, Berkeley, CA, pp 217–257

Baccouche A, Garcia-Zapirain B, Elmaghraby A (2018) Annotation Technique for Health-Related Tweets Sentiment Analysis. In: 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE, Louisville, KY, USA, pp 382–387

Badaro G, Baly R, Hajj H, et al (2014) A Large Scale Arabic Sentiment Lexicon for Arabic Opinion Mining. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP). Association for Computational Linguistics, Doha, Qatar, pp 165–173

Badaro G, El Jundi O, Khaddaj A, et al (2018) EMA at SemEval-2018 Task 1: Emotion Mining for Arabic. In: Proceedings of The 12th International Workshop on Semantic Evaluation. Association for Computational Linguistics, pp 236–244

Baly R, Badaro G, El-Khoury G, et al (2017a) A Characterization Study of Arabic Twitter Data with a Benchmarking for State-of-the-Art Opinion Mining Models. In: Proceedings of the Third Arabic Natural Language Processing Workshop. Association for Computational Linguistics, pp 110–118

Baly R, El-Khoury G, Moukalled R, et al (2017b) Comparative Evaluation of Sentiment Analysis Methods Across Arabic Dialects. Procedia Computer Science 117:266–273. https://doi.org/10.1016/j.procs.2017.10.118

Banea C, Mihalcea R, Wiebe J (2010) Multilingual Subjectivity: Are More Languages Better? In: Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference. pp 28–36

Baniata LH, Park S-B (2016) Sentence Representation Network for Arabic Sentiment Analysis. In: In Proceedings of The 43rd Annual Meeting and Winter Conference (제43회 정기총회 및 동계학술발표회). Gangwon-do , South Korea, pp 470–472

Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5:157–166. https://doi.org/10.1109/72.279181

Boudad N, Faizi R, Oulad Haj Thami R, Chiheb R (2018) Sentiment analysis in Arabic: A review of the literature. Ain Shams Engineering Journal 9:2479–2490. https://doi.org/10.1016/j.asej.2017.04.007

Brun C, Perez J, Roux C (2016) XRCE at SemEval-2016 Task 5: Feedbacked Ensemble Modeling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp 277–281

Cho K, van Merrienboer B, Gulcehre C, et al (2014) Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1724–1734

Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:14123555 [cs]

Cliche M (2017) BB\_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. arXiv:170406125 [cs, stat]

Cortes C, Vapnik V (1995) Support-vector networks. Machine Learning 20:273–297. https://doi.org/10.1007/BF00994018

Dahou A, Xiong S, Zhou J, et al (2016) Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. The COLING 2016 Organizing Committee, Osaka, Japan, pp 2418–2427

El-khair IA (2016) 1.5 billion words Arabic Corpus. arXiv:161104033 [cs]

El-Kilany A, Azzam A, El-Beltagy SR (2018) Using Deep Neural Networks for Extracting Sentiment Targets in Arabic Tweets. In: Shaalan K, Hassanien AE, Tolba F (eds) Intelligent Natural Language Processing: Trends and Applications. Springer International Publishing, Cham, pp 3–15

Elmadany AA, Mubarak H, Magdy W (2018) An Arabic Speech-Act and Sentiment Corpus of Tweets. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA)

Elman J (1990) Finding structure in time. Cognitive Science 14:179–211. https://doi.org/10.1016/0364-0213(90)90002-E

Elnagar A BRAD: Books Reviews in Arabic Dataset. In: https://github.com/elnagara/BRAD-Arabic-Dataset. https://github.com/elnagara/BRAD-Arabic-Dataset

Elnagar A, Lulu L, Einea O (2018) An Annotated Huge Dataset for Standard and Colloquial Arabic Reviews for Subjective Sentiment Analysis. Procedia Computer Science 142:182–189. https://doi.org/10.1016/j.procs.2018.10.474

González J-Á, Pla F, Hurtado L-F (2017) ELiRF-UPV at SemEval-2017 Task 4: Sentiment Analysis using Deep Learning. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, pp 723–727

Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, Vancouver, BC, Canada, pp 6645–6649

Habash NY (2010) Introduction to Arabic Natural Language Processing. Synthesis Lectures on Human Language Technologies 3:1–187. https://doi.org/10.2200/S00277ED1V01Y201008HLT010

Heckman S, Williams L (2011) A Systematic Literature Review of Actionable Alert Identification Techniques for Automated Static Code Analysis. Inf Softw Technol 53:363–387. https://doi.org/10.1016/j.infsof.2010.12.007

Heikal M, Torki M, El-Makky N (2018) Sentiment Analysis of Arabic Tweets using Deep Learning. Procedia Computer Science 142:114–122. https://doi.org/10.1016/j.procs.2018.10.466

Hemmatian F, Sohrabi MK (2017) A survey on classification techniques for opinion mining and sentiment analysis. Artificial Intelligence Review. https://doi.org/10.1007/s10462-017-9599-6

Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. Neural Computation 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Irsoy O, Cardie C (2014) Opinion Mining with Deep Recurrent Neural Networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 720–728

Kemp S (2018) Digital In 2018: Essential Insights into Internet, Social Media, Mobile, and Ecommerce Use Around The World

Kim Y (2014) Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1746–1751

Kiritchenko S, Mohammad S, Salameh M (2016) SemEval-2016 Task 7: Determining Sentiment Intensity of English and Arabic Phrases. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp 42–51

Kitchenham B (2004) Procedures for Performing Systematic Reviews. Keele, UK, Keele Univ 33:

Kobayashi T, Hirose K, Nakamura S (eds) (2010) Recurrent neural network based language model. ISCA, Makuhari, Chiba, Japan, pp 1045--1048

Kumar A, Kohail S, Kumar A, et al (2016) IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp 1129–1135

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539

Liu B, Zhang L (2012) A Survey of Opinion Mining and Sentiment Analysis. In: Aggarwal CC, Zhai C (eds) Mining Text Data. Springer US, Boston, MA, pp 415–463

Madhoushi Z, Hamdan AR, Zainudin S (2015) Sentiment analysis techniques in recent works. In: 2015 Science and Information Conference (SAI). pp 288–291

Medhat W, Hassan A, Korashy H (2014) Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal 5:1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient Estimation of Word Representations in Vector Space. arXiv:13013781 [cs]

Mohammad S, Bravo-Marquez F, Salameh M, Kiritchenko S (2018) SemEval-2018 Task 1: Affect in Tweets. In: Proceedings of The 12th International Workshop on Semantic Evaluation. Association for Computational Linguistics, pp 1–17

Mohammad SM, Salameh M, Kiritchenko S (2016) How Translation Alters Sentiment. Journal of Artificial Intelligence Research 55:95–130. https://doi.org/10.1613/jair.4787

Mourad A, Darwish K (2013) Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs. In: Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. Association for Computational Linguistics, pp 55–64

Nabil M, Aly M, Atiya A (2015) ASTD: Arabic Sentiment Tweets Dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pp 2515–2519

Olah C Understanding LSTM Networks. In: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Pang B, Lee L (2008) Opinion Mining and Sentiment Analysis. Foundations and Trends® in Information Retrieval 2:1–135. https://doi.org/10.1561/1500000011

Pang B, Lee L, Vaithyanathan S (2002) Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP ’02. Association for Computational Linguistics, Not Known, pp 79–86

Pascanu R, Mikolov T, Bengio Y (2013) On the Difficulty of Training Recurrent Neural Networks. In: Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28. JMLR.org, pp III–1310–III–1318

Pennington J, Socher R, Manning C (2014) Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1532–1543

Ponti EM, Vulić I, Korhonen A (2017) Decoding Sentiment from Distributed Representations of Sentences. In: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (\*SEM 2017). Association for Computational Linguistics, Vancouver, Canada, pp 22–32

Pontiki M, Galanis D, Papageorgiou H, et al SemEval-2016 Task 5 data and tools. http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools

Pontiki M, Galanis D, Papageorgiou H, et al (2016) SemEval-2016 task 5 : aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, pp 19–30

Rani S, Kumar P (2019) A journey of Indian languages over sentiment analysis: a systematic review. Artificial Intelligence Review 52:1415–1462. https://doi.org/10.1007/s10462-018-9670-y

Refaee E, Rieser V (2014) An Arabic Twitter Corpus for Subjectivity and Sentiment Analysis. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14). European Language Resources Association (ELRA), Reykjavik, Iceland, pp 2268–2273

Reinsel D, Gantz J, Rydning J (2018) The Digitization of the World From Edge to Core

Rosenthal S, Farra N, Nakov P (2017) SemEval-2017 task 4: Sentiment analysis in Twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp 502–518

Ruder S, Ghaffari P, Breslin JG (2016a) A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. arXiv:160902745 [cs]

Ruder S, Ghaffari P, Breslin JG (2016b) INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp 330–336

Ruder S, Ghaffari P, Breslin JG (2016c) A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, pp 999–1005

Saleh IM (2009) Automatic extraction of lemma-based bilingual dictionaries for morphologically rich languages

Samy AE, El-Beltagy SR, Hassanien E (2018) A Context Integrated Model for Multi-label Emotion Detection. Procedia Computer Science 142:61–71. https://doi.org/10.1016/j.procs.2018.10.461

Schmidhuber J (2015) Deep learning in neural networks: An overview. Neural Networks 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45:2673–2681. https://doi.org/10.1109/78.650093

Socher R, Pennington J, Huang EH, et al (2011) Semi-supervised Recursive Autoencoders for Predicting Sentiment Distributions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 151–161

Socher R, Perelygin A, Wu J, et al (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Seattle, Washington, USA, pp 1631–1642

Soliman AB, Eissa K, El-Beltagy SR (2017) AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. Procedia Computer Science 117:256–265. https://doi.org/10.1016/j.procs.2017.10.117

Souri A, El Maazouzi Z, Al Achhab M, El Mohajir BE (2018) Arabic Text Generation Using Recurrent Neural Networks. In: Tabii Y, Lazaar M, Al Achhab M, Enneya N (eds) Big Data, Cloud and Applications. Springer International Publishing, Cham, pp 523–533

Tai KS, Socher R, Manning CD (2015) Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Beijing, China, pp 1556–1566

Tamchyna A, Veselovská K (2016) UFAL at SemEval-2016 Task 5: Recurrent Neural Networks for Sentence Classification. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, pp 367–371

Tang D, Qin B, Liu T (2015) Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pp 1422–1432

Turki Khemakhem I, Jamoussi S, Ben Hamadou A (2010) Arabic morpho-syntactic feature disambiguation in a translation context. In: Proceedings of the 4th Workshop on Syntax and Structure in Statistical Translation. Coling 2010 Organizing Committee, Beijing, China, pp 61–65

Vohra MSM, Teraiya J (2013) A COMPARATIVE STUDY OF SENTIMENT ANALYSIS TECHNIQUES 1

Wang B, Lu W (2018) Learning Latent Opinions for Aspect-level Sentiment Classification. In: AAAI

Wang P, Qian Y, Soong FK, et al (2015) A Unified Tagging Solution: Bidirectional LSTM Recurrent Neural Network with Word Embedding. arXiv:151100215 [cs]

Yadav A, Vishwakarma DK (2019) Sentiment analysis using deep learning architectures: a review. Artificial Intelligence Review. https://doi.org/10.1007/s10462-019-09794-5

Young T, Hazarika D, Poria S, Cambria E (2018) Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. IEEE Computational Intelligence Magazine 13:55–75. https://doi.org/10.1109/MCI.2018.2840738

Zhang L, Wang S, Liu B (2018) Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8:e1253. https://doi.org/10.1002/widm.1253

Zhang Y, Jin R, Zhou Z-H (2010) Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics 1:43–52. https://doi.org/10.1007/s13042-010-0001-0

1. https://github.com/bakrianoo/aravec [↑](#footnote-ref-1)
2. https://github.com/iamaziz/ar-embeddings [↑](#footnote-ref-2)
3. https://fasttext.cc [↑](#footnote-ref-3)