Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation

Alayba, A. M. & Palade, V

Original citation:
Alayba, AM & Palade, V 2021, 'Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation', Journal of King Saud University - Computer and Information Sciences.
https://doi.org/10.1016/j.jksuci.2021.12.004

DOI 10.1016/j.jksuci.2021.12.004
ISSN 1319-1578

Publisher: Elsevier

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation

Abdulaziz M. Alayba [a,*], Vasile Palade [b]

[a] Department of Information and Computer Science, College of Computer Science and Engineering, University of Ha'il, Ha'il 81481, Saudi Arabia
[b] Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry CV1 5FB, UK

ARTICLE INFO

Article history:
Received 24 August 2021
Revised 14 November 2021
Accepted 6 December 2021
Available online xxxx

Keywords:
Arabic NLP
Arabic sentiment analysis
CNN
LSTM
Word embedding for Arabic
Arabic word normalisation
Deep learning

ABSTRACT

The high variety in the forms of the Arabic words creates significant complexity related challenges in Natural Language Processing (NLP) tasks for Arabic text. These challenges can be dealt with by using different techniques for semantic representation, such as word embedding methods. In addition, approaches for reducing the diversity in Arabic morphologies can also be employed, for example using appropriate word normalisation for Arabic texts. Deep learning has proven to be very popular in solving different NLP tasks in recent years as well. This paper proposes an approach that combines Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks to improve sentiment classification, by excluding the max-pooling layer from the CNN. This layer reduces the length of generated feature vectors after convolving the filters on the input data. As such, the LSTM networks will receive well-captured vectors from the feature maps. In addition, the paper investigated different effective approaches for preparing and representing the text features in order to increase the accuracy of Arabic sentiment classification.

1. Introduction

Sentiment analysis is one popular NLP task that aims to determine the feeling from a certain text (Dey et al., Jul. 2016). It has been well explored by many researchers in many languages including Arabic. A lot of research has been conducted on sentiment analysis in different domains, such as politics (Elghazaly et al., 2016), health services (Alayba et al., 2017), cybersecurity (Al-Rowaily et al., 2015), costumer reviews about service companies (Almuqren and Cristea, 2016), and economy (Yang et al., 2020).

The customer opinions represent a valuable source of data, but extracting these opinions from unstructured text is a very challenging task (Adnan and Akbar, 2019). This is usually done using machine learning techniques and different text representation methods. In order to perform the sentiment classification, researchers have employed basic machine learning algorithms and considered different feature selection techniques. Many feature extraction and selection methods have been used, such as bag-of-words, part-of-speech (POS) tagging, Term Frequency (TF) and Term Frequency Inverse Document Frequency (TF-IDF) (El-Din, 2016). Several machine learning algorithms have been employed for sentiment classification in Arabic. For instance, Naïve Bayes (NB), Support Vector Machine (SVM), Decision-Tree, K-Nearest Neighbours, and Logistic Regression have been used in (Itani et al., 2012). In the past five years, most of the researchers in the area showed a tendency towards investigating the performance of deep learning algorithms with word embedding for sentiment classification tasks. Four deep learning models were proposed for sentiment analysis in (Al Sallab et al., 2015), including Deep Neural Networks (DNN), Deep Belief Networks (DBN), Deep Auto Encoders (DAE), and Recursive Auto Encoders (RAE). However, the input data were represented using traditional techniques such as bag-of-words and using a sentiment score lexicon. Several Arabic Word2Vec models were built and then used to feed a CNN model for sentiment classification in (Dahou et al., 2016). The sentiment model was evaluated using some Arabic datasets already available. The research of (Altowayan and Elnagar, 2017) employed continuous bag-of-words (CBOW), fastText and Skip-gram fastText models for Arabic with 100 dimensions, and used them with five different classifiers. The BiLSTM method was used for sentiment

[*] Corresponding author.
E-mail address: a.alayba@uoh.edu.sa (A.M. Alayba).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.12.004
1319-1578/© 2021 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

analysis with integrated word vector representation and TF-IDF in (Xu et al., 2019). Other models using deep learning will be detailed in the next section.

This paper introduces a state-of-the-art model for Arabic sentiment classification, where a CNN and an LSTM network are concatenated by eliminating the pooling layer of the CNN. We eliminated the Max-pooling layer from the CNN with a view to presenting the features effectively. The size of the convolving filters is fixed and the resulting vectors in the feature maps feed into the LSTM cells as inputs. We compare this approach with our previous Word-Level based model in (Alayba et al., 2018) and we revealed that the new model have better results, as shown in Section 6 Table 6. In addition, we investigated two approaches for feature representation, which are: word normalisation and word embedding. For word normalisation, we used various techniques to pre-process the Arabic words, i.e.: MADAMIRA (Pasha et al., 2014), Farasa (Abdelali et al., 2016) and Stanford (Manning et al., 2014). In the word embedding case, we compared three different approaches on the same corpora: Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014) and fastText (Bojanowski et al., Dec. 2017), with three different dimensions, i.e. 100, 200 and 300.

The rest of the paper is organised as follows. Some related works on the use of deep learning, especially for NLP, are described in Section 2. Section 3 illustrates the four Arabic sentiment analysis datasets used to evaluate the proposed model and feature processing techniques. Section 4 elaborates on all of the details of pre-processing the Arabic text in order to represent the features effectively. The CNN-LSTM networks for the sentiment classification approach is detailed in Section 5. In Section 6, the results obtained by the proposed sentiment classification model are discussed and compared with results obtained by other models. Section 7 highlights the main conclusions of the experiments in this paper and outlines future work.

2. Related work

Deep learning has proven to be an excellent tool for solving a lot of challenging problems in the past years. It has been used in variety of applications, e.g. self-driving cars (Rao and Frtunikj, 2018), digital marketing (Ribeiro et al., 2017), image colourisation (Hwang and Zhou, 2016), visual recognition (Liu et al., Sep. 2019), speech recognition (Afouras et al., 2018), fraud detection (Roy et al., 2018), disasters prediction (Aqib et al., 2018), etc. In addition, deep neural networks have been employed in the natural language processing field by many researchers, for example, for machine translation (Mahata et al., 2019), chatbot customer service (Xu et al., 2017), question answering (Ribeiro et al., 2018), automatic summarisation (Merchant and Pande, 2018), spelling correction (Etoori et al., 2018). The related work in this paper will discuss recent and effective works in the field of sentiment analyses for Arabic language as well as other languages. It is divided into two sub-sections; the first sub-section will explore sentiment analysis for other languages, and the second one is on sentiment analysis for Arabic.

2.1. Sentiment analysis for other languages

There is a huge number of studies on different aspects of sentiment analysis for many languages, and especially for English, which makes the majority of papers in this area. Sentiment analysis became an active research area in the first years of this century, and the interest undoubtedly expanded after the appearance of social media around 2006 (Liu, 2015). Different feature selection and extraction methods have been used in these studies, such as words' appearance and frequency, N-gram (Ahuja et al., 2019), POS (Fang and Zhan, 2015), lexicon-based (Khoo and Johnkhan, 2018), contextual entropy model and Principal Component Analysis (Yu et al., 2013), Latent Semantic Indexing (Rizun et al., 2018), and others. Supervised learning approaches were employed for sentiment analysis (Vilares et al., 2017), as well as unsupervised learning techniques in (Cheng et al., 2017), semi-supervised learning in (Khan et al., Jun. 2017), and in general multiple other basic machine learning methods, as used in (Neethu and Rajasree, 2013; Agarwal and Mittal, 2016; Rahman and Hossen, 2019), and (Jagdale et al., 2019).

From 2011, this field of research tended to employ deep learning models, and (Socher et al., 2011) proposed Recursive Autoencoders Network (RAE), which seek for a reduced dimensional vector representation for sentiment classification. Furthermore, the work of (Socher et al., 2013) developed a Recursive Neural Tensor Network (RNTN) model, where the tensor is used for capturing better sentiment features. A dynamic CNN (DCNN) was introduced for question and sentiment analysis in (Kalchbrenner et al., 2014), and this model used dynamic k-max pooling as a non-linear subsampling function. Four different variants of CNN models were built for sentence classification (Kim, Sep. 2014), namely, CNN-rand, CNN-static, CNN-non-static, and CNN-multichannel. A Character to Sentence CNN (CharSCNN) model, which contained two convolutional layers to capture the features from words and sentences, was employed to measure the sentiment analysis for short text (dos Santos and Gatti, 2014). A lexicon embedding integrated with a CNN model improved the sentiment classification, compared with the CNN model only (Shin et al., 2017). The researchers of (Wang et al., 2015) captured the sentiment using an LSTM model by simulating the occurrences of words through a compositional function, and a bidirectional LSTM model was used with attention mechanisms to empower the network for sentiment analysis. A joint convolution and recursive neural network model was proposed for English sentiment classification using Word2Vec (Mikolov et al., 2013), which used a recursive neural network instead of the max-pooling layer (Sadr et al., 2019). Also, the work of (Van et al., 2017) integrated a convolutional layer with recurrent and recursive neural networks for sentiment analysis using Glove (Pennington et al., 2014). Another combining model, which contains two parts for sentiment analysis, was presented in (Wang et al., 2016), and it used a regional CNN and an LSTM to measure the rate of valence arousal in the text. The research of (Wang et al., 2016) presented a joint CNN and Recurrent Neural Network (RNN) for short text sentiment analysis in order to benefit of the advantage of using both the local and global features.

2.2. Sentiment analysis for Arabic language

The interest in building deep learning models for Arabic NLP has also grown in recent years. The work of (Abdullah and Shaikh, 2018) introduced a deep learning system to reveal emotions in English and Arabic tweets. Sentence representation was applied to a deep neural machine translation model to translate the text from English to four other languages, of which Arabic was one (Poliak et al., 2018). The research of (Al-Smadi et al., 2018) compared deep recurrent neural network approaches for Arabic sentiment analysis with a support vector machine. The pre-trained word embedding techniques for Arabic sentiment analysis were employed on tweets about airlines (Ashi et al., 2019, 2018). The work of (Soliman et al., 2017) built AraVec, which is a set of six word embedding models for Arabic NLP tasks. A hybrid CNN and LSTM model was proposed for Arabic sentiment analysis in (Al-Azani and El-Alfy, 2017), where the model was trained using the word2vec technique. The role of using word level, character level and five-character level for Arabic text were examined (Alayba

Table 1
Related Sentiment Analysis Approaches Comparison.

Sentiment analysis approaches	Important characteristics	Cons
(Alayba et al., 2018)	A combined CNN and LSTM model	Used five-character level, which did not represent the features well
(Ahuja et al., 2019)	The impact of features extraction on the sentiment analysis	Used some basic features extraction only and some basic machine learning algorithms only
(Rahman and Hossen, 2019)	A novel movie review dataset was proposed	Used some basic machine learning algorithms only
(Jagdale et al., 2019)	A novel product reviews dataset was proposed	Used some basic machine learning algorithms only
(dos Santos and Gatti, 2014)	CNN from character- to sentence level	The filter can capture different words from the sequence of characters
(Sadr et al., Dec. 2019)	A combined CNN and Recursive Neural Networks	Used Word2Vec (Mikolov et al., 2013) Skip-Gram only based on the results of other paper
(Abdullah and Shaikh, 2018)	Emotion detection using deep learning	Part of the inputs are translated from Arabic to English using a tool
(Ashi et al., 2019, 2018)	Covered different aspects of the reviews in Arabic	Both word embedding models pre-trained on different corpora – unfair comparison
(Al-Azani and El-Alfy, 2017)	A hybrid CNN and LSTM model	Used Word2Vec (Mikolov et al., 2013) only with different techniques (namely CBOW and SG)
(Heikal et al., 2018)	A combined CNN and LSTM model	The paper did not consider any feature engineering task
(Al Omari et al., 2019)	A hybrid CNN and LSTM model	The number of epochs is small, and it is not satisfying for some experiments
(Kaibi et al., 2020)	Concatenating two different word embedding models	Using simple classifiers: Linear SVC, Random Forest, Gaussian Naive Bayes, NuSVC, Logistic Regression, and Stochastic Gradient Descent Classifier

et al., 2018). These features were used for sentiment analysis purposes and employing deep neural network architectures (CNNs and LSTMs). The sentiment analysis on the Arabic dataset called ASTD (Nabil et al., 2015), by using a joint CNN and LSTM model, was developed in (Heikal et al., 2018). An intensive CNN-LSTM model by doubling the Convolutional and Max-pooling layers was implemented to measure the accuracy of classifying five different Arabic sentiment datasets in (Al Omari et al., 2019). A deep learning model by merging Bidirectional LSTM with CNN, where the features were represented using AraVec (Soliman et al., 2017), was presented in (Abu Kwaik et al., 2019) for Arabic binary classification. The research of (Baly et al., 2017) considered the diversity of Arabic morphology using the Arabic sentiment treebank (ARSENTB), and they used it with Recursive Neural Tensor Networks (RNTN). The sentiment classification using three different datasets was examined in (Al-Sallab et al., 2017) using a deep learning model named Recursive Deep Learning Model for Opinion Mining in Arabic (AROMA). The work of (El-Kilany et al., 2018) detected the sentiment targets in Arabic tweets using word embedding techniques as input for Bidirectional LSTMs, together with a Conditional Random Field (CRF) layer. (Table 1) shows several related sentiment analysis approaches comparison.

3. Datasets

There is very limited availability of good Arabic sentiment datasets. In this paper, we only consider the Arabic datasets with binary classes because we employ a binary classification model. We used four Arabic sentiment datasets (where one of them was a subset of another one) in this paper, in order to test the performance of the proposed classification model and Arabic word preprocessing. We prepared each dataset in two txt files, where each one contains the reviews for one class. The datasets are as follows:

3.1. Arabic Health Services Dataset (Main-AHS and Sub-AHS)

We firstly presented the Main-AHS dataset in (Alayba et al., 2017), to be used for Arabic sentiment analysis purposes and other Arabic NLP tasks. It contains reviews about health services in the Arabic language, and it was collected from Twitter. Each review was annotated and classified by three Arabic speakers, as either positive or negative, and the majority voting was taken afterwards. That resulted in 1398 negative and 628 positive tweets, which makes a total of 2026 tweets. Furthermore, a sub dataset of the Main-AHS was introduced in (Alayba et al., 2018) and was named Sub-AHS. It has 502 positive and 1230 negative tweets, making a total of 1732 tweets. It contains all the tweets where all three annotators were in agreement, on either the positive or negative category. The tweets in these datasets were written in Saudi Arabian dialects.

3.2. Twitter Data Set (Ar-Twitter)

Ar-Twitter is an available Arabic sentiment dataset crawled from Twitter (Abdulla et al., 2013). The collected tweets focused on a variety of fields, i.e. arts, communities and politics. It was manually annotated into two classes as either positive or negative. The dataset contains 2000 tweets, where each class has 1000 tweets. However, due to some tweets being missing in the available online dataset, the negative class in our experiment has 975 tweets and the positive class 1000. The tweets in this dataset are written in either the Jordanian dialect or Modern Standard Arabic (MSA).

3.3. Arabic Sentiment Tweets Dataset (ASTD)

ASTD is another freely available dataset for sentiment analysis purposes in the Arabic language, which was first proposed in (Nabil et al., 2015). The dataset was collected from Twitter and the number of tweets before filtering was over 54,000. Then, the Amazon Mechanical Turk (AMT) service was used to label the tweets manually into four classes: objective, subjective positive, subjective negative, and subjective mixed. Due to the conflict in the rating results for most of the tweets, such tweets have been ignored and the final number of labelled tweets was 10,006. The model in this paper only considers two classes and, because of that, we will only utilise the positive and negative classes. The number of tweets that are used in this paper is 2479 tweets (1684 negative and 795 positive).

4. Features preparation

The richness of the morphology in the Arabic language complicates the NLP tasks compared with English. There are many forms of detached and attached pronouns in the Arabic language. The detached pronouns are separated from the word itself and each one is treated as a single token. However, the attached pronouns are linked to the end of the Arabic word, so it increases the forms of a single word. Moreover, there are many forms of prefixes and suffixes in Arabic for a single word. For example, the word "nominated" in Arabic رشح can have a multiple number of forms, such as ترشحونهما، أيرشحونهما، سترشحونهما، سيرشحونهما، فترشحونهما، فسترشحونهم، فسيرشحونهم، فيرشحونهما، وترشحونهما، وسترشحونهم، وسيرشحونهم، ويرشحونهما etc. Table 2 shows an example for the Arabic word أيرشحونهما, and it indicates the core of the word with its prefixes and suffixes.

The machine treats a word with different prefixes and suffixes differently, and for this matter there is a need to reduce the number of forms. There are several NLP techniques to deal with this issue, such as stemming, lemmatisation, and Arabic word segmentation. Stemming is an approach to remove the affixes in the words or delete the end of the word (Althobaiti et al., 2014). Lemmatisation is a procedure to consider the morphological analysis of the words and convert the words into their core or root form (Manning et al., 2008). Arabic word segmentation is a technique which splits prefixes and suffixes from the stem or the root of the word (Monroe et al., 2014). The subsection 4.1 provides more details and techniques that are used in this paper. Word embedding is an NLP technique that is capable of capturing the semantic and syntactic word based on the context of a word in a very large document (Bian et al., 2014). This technique has the ability to learn from a large text corpus and to represent a set of words in n-dimensional vectors of real numbers. Further details about word embedding are given in subsection 4.2.

4.1. Word normalisation

In this paper, we used three readily available tools to preprocess the Arabic text: MADAMIRA (Pasha et al., 2014), Farasa (Abdelali et al., 2016) and Stanford (Manning et al., 2014).

4.1.1. MADAMIRA

MADAMIRA is a morphological Arabic tool to apply some analytical techniques on Arabic text (Pasha et al., 2014). This system was derived from two other Arabic pre-processing systems, MADA (Nizar Habash et al., 2013) and AMIRA (Mona Diab and Hacioglu, 2007). It deals with two types of Arabic dialects as input text, which are either Modern Standard Arabic (MSA) or the Egyptian dialect. It has several NLP functionalities, which are Part-of-Speech, text Tokenisation, text Diacritisation, text Stemming, text Lemmatisation, Base Phrases, Glossary, and Named Entities. It is freely available online at https://camel.abudhabi.nyu.edu/madamira/.

The online version can only receive up to 200 characters with spaces as input at one time. Therefore, we downloaded the MADAMIRA java archive file, and we ran the following code for negative and positive text files for each dataset:

```
java -Xmx2500m -Xms2500m -XX:NewRatio = 3 \\
-jar<location of the MADAMIRA.jar file> \\
-rawinput inputData.txt \\
-rawoutdir<output directory> \\
-rawconfig rawConfig.xml
```

Each output file is in XML format and it contains a lot of information. The information is the output of all the previously mentioned functionalities and other XML tags. Consequently, there is a need to parse the output XML file to extract the required and useful information. We only considered the segmented text, lemmatised text, and stemmed text. We used the Element Tree package (Foundation, 2020) in Python (Foundation, 2020) to parse the output XML files to extract the segmented, lemmatised and stemmed lines. The MADAMIRA technique used the "+" sign to mark the split word, where the affixes divide from the stem word form.

This was used in the segmentation and lemmatisation stages. Moreover, the outputs of the lemmatisation and stemming contain diacritics (short vowels). Table 3 shows examples for the Segmentation, Lemmatisation, and Stemming outputs. We removed the "+" signs from the segmented and lemmatised text. Also, we eliminated the diacritics from the lemmatised and stemmed text.

4.1.2. Farasa

Another Arabic morphological and text processing toolkit that we used was Farasa (Abdelali et al., 2016). It has eight different Arabic NLP tasks, which are: text segmentation, spell checking, part-of-speech tagging, text lemmatisation, text discretisation, dependency parsing, constituency parsing, and name entity recognising. It has a demo available online at: http://qats-demo.cloudapp.net/farasa/, and the length of the input text should not exceed 400 characters. Alternatively, a Java archive file can be downloaded, and there is a separate file for each Farasa function. We only applied the Farasa segmentation and lemmatiation functions for this study. Hence, we ran Algorithm 1 on Java using the downloaded Farasa archive file for segmentation and lemmatisation.

Algorithm 1: Applying Farasa segmentation and lemmatisation on the datasets

```
Input: input text file;
Output: output text file;
While line != null do
  Segment line;
  For segmented line do
    Write segmented token into Output;
  End For
End While
```

Algorithm 1 was used to segment the Arabic text and, in order to do the lemmatisation, we used the lemmatise function instead of the segment function. The outputs of the lemmatisation function do not need any further preprocessing. The output of the segmentation function uses the "+" sign to split the affixes from the stem form of

Table 2
An Example of An Arabic Word with Prefix and Suffix.

Prefix		Core	Suffix	
Antefix	Prefix		Suffix	Postfix
أ	ي	رشح	ون	هما
A letter indicates a Yes or No question	A letter indicates the present tense for a masculine person	Most of the Arabic words have a core consisting of three letters and this word means "Nominated"	Termination for masculine plural	A pronoun for masculine dual, which means "them"

Table 3
An example of an Arabic text input and the outputs for Madamira approaches.

The MADAMIRA approaches	The output
Original (input line)	خدمه الاسعاف الطائر خدمه جميله لاسعاف حالات في وقت وجيز
Segmentation	خدم +ه ال+ اسعاف ال+ طائر خدم +ه جميل +ه ل+ اسعاف حالات في وقت وجيز
Lemmatisation	خَدَم +هُ إِسْعاف طائِر خَدَم +هُ جَمِيل +هُ لِ+ إِسْعاف حالَة فِي وَقْت وَجِيز

Table 4
An example of an Arabic text input and the outputs for Farasa approaches.

The Farasa approaches	The output
Original (input line)	خدمه الاسعاف الطائر خدمه جميله لاسعاف حالات في وقت وجيز
Segmentation	خدم+ه ال+اسعاف ال+طائر خدم+ه جميل+ه ل+اسعاف حال+ات في وقت وجيز
Lemmatisation	خدم إسعاف طائر خدم جميل اسعاف حالة في وقت وجيز

the word. Thus, we replaced the "+" sign by a space " ", with a view to separate the affix tokens and the stem form tokens. Table 4 shows examples of the Farasa segmentation and lemmatisation outputs.

4.1.3. Stanford

Stanford University developed the Stanford CoreNLP (Manning et al., 2014), an integrated software for NLP tasks. This toolkit supports the text pre-processing for different human languages, e.g., English, Chinese, Spanish, Arabic, etc. The functionalities for each language are grouped in separate packages and each package provides several text annotation tasks. The Arabic package can do word segmentation, sentence splitting, part-of-speech tagging, and constituency parsing. This toolkit is built in Java and it can be run as a server. The following code is to run the JAR file server of StnfordCoreNLP for Arabic:

```
    java -Xmx4g -cp ''*"
    edu.stanford.nlp.pipeline.StanfordCoreNLPServer  -
serverProperties
    StanfordCoreNLP-arabic.properties  -port  9000  -
timeout 1500
```

After running the server, we can make the API calls to use any functionality in the package. There are four output formats of the API for the annotated property: JSON, XML, Text, and Serialised. The tokenize annotation function was used to segment the Arabic text and the output was formatted in JSON. Algorithm 2 was written in Python in order to segment the Arabic sentiment datasets. Table 5 presents an example of the Stanford segmentation outputs with the original text input.

Algorithm 2: Applying Stanford segmentation on the datasets

```
Create a connection to CoreNLPServer;
Input: input text;
Output: output text;
While line != null do
  segment line into JSON format;
  For segmented word in JSON do
    Append segmented word into an array;
  End For
  For elements in array do
    Write elements into Output;
    write " " into Output;
  End For
End While
```

4.2. Word Embedding

Word embeddings are distributed word vector representations, where words with similar meanings have similar vector representations (Chen et al., 2015). The size of the vector is fixed, and the words are distributed in very high dimensional space. A very large corpus of text is fed to the model as input and based on that the words are distributed in vectors. There are many approaches to word embeddings, such as Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), fastText (Bojanowski et al., Dec. 2017), Elmo (Peters et al., 2018) and Poincaré Embeddings (Nickel and Kiela, 2017). Each technique considers different factors to represent the words. The input row of text has the main role in the word representation, where the context of words can change the word distribution. In this work, we used a readily available large Arabic corpus, which contains over 1.5 billion words called the Abu El-Khair Corpus (Abu El-khair, 2016). We already filtered this corpus in (Alayba et al., 2018) to be used for word embedding. W considered only three word embedding techniques in this paper: Word2-Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014) and fastText (Bojanowski et al., Dec. 2017).

4.2.1. Word2Vec Model

The Word2Vec method was introduced in (Mikolov et al., 2013) and it used neural network techniques to obtain word representations. The method takes a large corpus as input and considers all the vocabulary in this corpus. This method's idea is that words have similar meanings occur in similar contexts (Harris, 1954). The algorithm updates the vectors of the words based on the appearance of this word within the surrounding context using a fixed size window. The similarity between these words is increased and the vectors will be convergent. Word2Vec has two techniques to generate the word vectors, which are Continuous Bag-of-Words (CBOW) and Skip-gram (SG). Fig. 1 highlights the difference between the two approaches. In the CBOW, the vector of the centre word is updated based on the surrounding context within the

Table 5
An example of an Arabic text input and the output for Stanford approach.

The Farasa approaches	The output
Original (input line)	خدمه الاسعاف الطائر خدمه جميله لاسعاف حالات في وقت وجيز
Segmentation	خدم ه الاسعاف الطائر خدم ه جميل ه ل اسعاف حالات في وقت وجيز

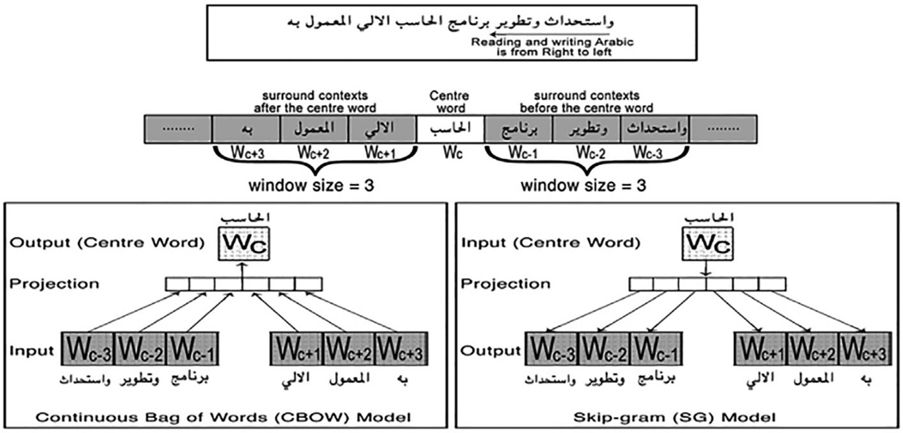

Fig. 1. Continuous Bag of Words (CBOW) and Skip- Gram (SG) Models for Word2Vec (Alayba et al., 2018).

window size. Whereas in the SG model, the vectors of the surrounding context within the window size are changed based on the centre word. We used a window size of five for all the models, and the minimum count of words was equal to five. Also, we used three different dimensions for both techniques, which were: 100, 200 and 300.

4.2.2. Glove Model

GloVe is another word embedding technique, which was proposed in (Pennington et al., 2014). This approach uses an unsupervised learning method to build word embedding vectors in a space. The aims of this model are similar to Word2Vec in terms of clustering similar words and repelling different words. However, the mechanism of this technique is different from Word2Vec. GloVe not only considers the context of the word, which is the surrounding words, but it also examines the occurrences of all the words in the corpus. Therefore, both local and global statistics in the corpus are needed in this model to distribute the word vectors. This model focuses on the non-zero values in a global word to word co-occurrences matrix. It measures the ratio of co-occurrence probabilities of these two words together from the input corpus. The affinity of these words can be disclosed if the ratio is large, and vice-versa. In this paper, we constructed three different dimensions for both techniques: 100, 200 and 300. Also, we used a window size of five and the minimum appearance of the word in the corpus was equal to five as well.

4.2.3. fastText Model

Another word embedding technique was presented in (Bojanowski et al., 2017), which is called fastText. It is based on an unsupervised algorithm to represent words as vectors. It is an extension of the Word2Vec model, where it takes into consideration the sub-words. Also, it has two architectural models that are Continuous Bag-of-Words (CBOW) and Skip-gram (SG). However, fastText subdivides each word into an n-gram character. It uses angular brackets as a special boundary as an indication of the beginning and end of the word. It is for differentiating from a word itself and a sub-word from another word. For instance, the fastText representation for the word "sentiment" when n = 4 is <sen, sent, enti, ntim, time, ment, ent>. The sequences <sent> and <time> refer to the words sent and time, which are different from the 4-gram sent and time from the word sentiment. Considering sub-words helps this model to distinguish between the prefixes and suffixes as well as shorter character sequences of the word. This model includes the word itself to be represented in a vector with the set of its character n-grams. The sub-words are linked to their original word in a hashable list and the sum of the n-gram vectors is equal to the vector of the original word. We also used the same dimensions as in previous models, which were: 100, 200 and 300. Also, we employed the same window size and minimum word count to make a fair comparison.

All the Arabic Word Embedding models are freely available for researchers only in the following link https://zenodo.org/record/4739760.

5. Methodology

We proposed a combined CNN-LSTM model to do the sentiment classification for Arabic text in our previous work (Alayba et al., 2018). However, in this paper we developed the model to improve the sentiment classification using a range of different techniques. We used different Arabic normalisation methods and different word embedding techniques in the input layer. Also, we increased the number of convolving filters in the convolutional layer to gain different features. Additionally, we boosted the number of LSTM cells to fit with the output of the convolutional layer. In this model, we omitted the Max-pooling layer to avoid any absence of features before feeding them into the LSTM layer. Fig. 2 shows the architecture of the sentiment classification model, which was built using the Keras tool (dos Santos and Gatti, 2014; Chollet, et al., 2020).

5.1. Input layer

This layer contains the row of vectors, and each vector represents a token of text. For the word normalisation part, the token can be a whole word, a suffix, a prefix, the root of a word, or the core of a word. The vectors of each token in this part are represented without pre-training, because the words in the available Abu El-Khair Corpus are not normalised. The number of vectors was equal to the longest review for the input dataset. Therefore, all the reviews were padded using the token<Pad> to the length of the longest review, in order to have the same size matrices for each review. The length of the vectors was fixed to 200.

For the word embedding approach, we used the original datasets, i.e., word level without any word processing. Also, we used three different vector lengths, which were: 100, 200 and 300. Moreover, we used three different techniques of pre-trained vectors: Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014) and fastText (Bojanowski et al., 2017), as mentioned in Subsection 4.2. The input layer is represented as a matrix with the form ($\ell \times d$), where ℓ is the number of tokens in the tweets (vectors) and d is the length of the token vector.

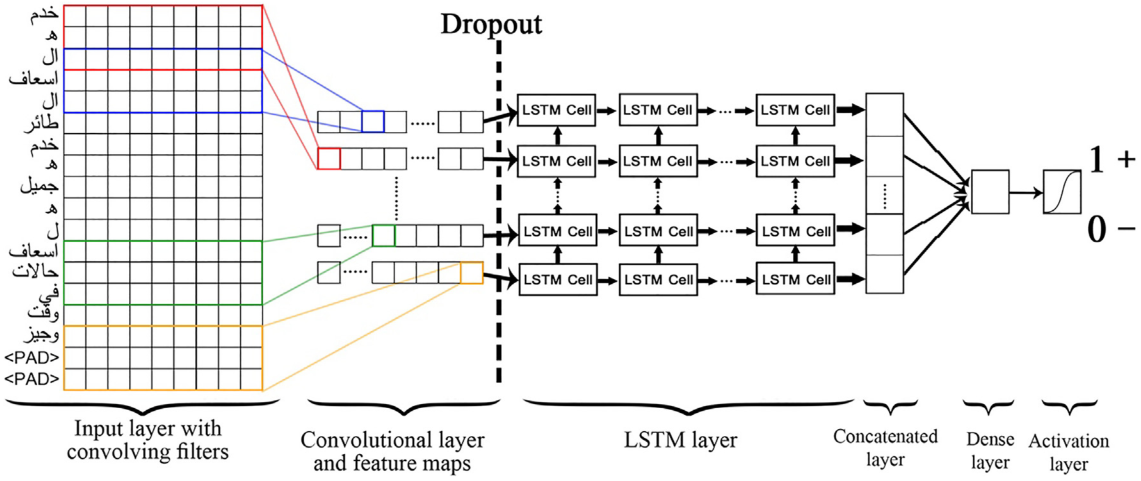

Fig. 2. An enhanced CNN and LSTM approach for Arabic sentiment classification.

5.2. Convolutional layer

The length of convolving filters (kernel size) was assigned to 3 for all the experiments in this paper. The number of filters (kernels) were determined to be ($\ell - 3$), where ℓ is the number of tokens and 3 is the size of the sliding filter. The features map $\boldsymbol{M} \in \mathbb{R}^{\ell \times d}$, where ℓ is the number of tokens, and d is the dimensionality of token vectors. Each filter slides over the input layer to extract features from the represented vectors for each review. The filter starts sliding from the beginning of the layer to the end. In each sliding step, the filter generates the best features from three tokens using the activation function Rectified Linear Unit (ReLU) (Nair and Vinod and Hinton, 2010). Then, these features were concatenated in feature maps as vectors. For example, in Fig. 2, each feature map is a vector, and it is a result of convolving filters. Therefore, the number of feature maps is equal to the number of convolving filters ($\ell - 3$). A single feature in the feature map m_i is generated from a convolving filter of ℓ tokens $x_{i:i+\ell-1}$ by $m_i = f(w.x_{i:i+\ell-1} + b)$. Where, $b \in \mathbb{R}$ is a bias term and f is a non-linear function. As a result, the main role of this layer is to nominate the valuable features from the input layer based on the used activation function and create them in the feature map.

5.3. Dropout layer

After the convolutional layer, each feature map vector passes through the dropout layer to prevent overfitting of the neural network. This layer provides a technique to regularise this deep learning model. Also, it improves the generalisation techniques for the network to equally consider all the inputs in the LSTM layers without focusing on a specific one. Thus, this layer avoids any biases in the training of these deep neural networks.

5.4. LSTM layer

After regularising the vectors in the dropout layer, each vector is fed to an LSTM cell. The number of LSTM units are set to the number of filters in the convolutional layer multiplied by 3. These cells are the main components of the LSTM layer, which is a feedback neural network. It can process both single data and sequential data and it shows good performance on sequence data, such as speech, video and a row of text (Baytas et al., 2017). In this model, we predicted the class for a row of text, and it was important to consider the context of the text or the sequential features. This layer uses the back-propagation techniques with a view to improve the text classification.

5.5. Dense and activation layers

The role of the dense layer is to integrate the outputs of LSTMs together as a single vector. Then, it is also responsible for computing the concatenated vector to a single value, which ranges from 0 to 1. The single yield value from the dense layer is passed through the activation layer to determine the text into its class, either positive or negative. The sigmoid function (Han and Moraga, 1995) was used in the activation layer in this model. It outputs between 0 and 1 to predict the class probability for the input text.

6. Results and analysis

The proposed model merges two different neural networks, which are CNN and LSTM, by using Keras as a development tool (dos Santos and Gatti, 2014; Chollet, et al., 2020). In this model, we dropped the Max-pooling layer in the CNN and we also intensified the number of convolving filters and the number of LSTM cells compared with our previous model in (Alayba et al., 2018). We divided all the datasets into 80% for training and the remaining 20% for testing the model in all the experiments. Moreover, we ran each experiment more than once, and then the average accuracy was measured for each one. We tested the classification performance using the accuracy over 50 epochs for each experiment, measured using the following well-known relation (Manning et al., 2008).

$$ACC = (TP + TN) \;/\; (TP + TN + FP + FN)$$

Here, *TP* is the number of positive reviews that are correctly predicted as positive, *TN* is the number of negative reviews that are correctly predicted as negative, *FP* is the number of negative reviews that are incorrectly predicted as positive, and *FN* is the number of positive reviews that are incorrectly predicted as negative.

At the beginning, we examined this model using the same word level features, which we had previously used in (Alayba et al., 2018). There are slight improvements in the accuracy of the results for Sub-AHS and Ar-Twitter datasets. Moreover, there are clear increments in the accuracy result for the ASTD dataset in the new models. Table 6 compares the accuracy results of both models based on the four datasets.

Table 6
Accuracy comparison of the proposed model (CNN + LSTM without Max-pooling) with the previous model in (Roy et al., 2018) (CNN + LSTM with Max-pooling) based on word-level for The Same Datasets.

	Main-AHS	Sub-AHS	Ar-Twitter	ASTD
Previous Model (Alayba et al., 2018)	**0.9424**	0.9510	0.8810	0.7641
Proposed Model	0.9335	**0.9539**	**0.8861**	**0.7823**

1) The emboldened and underlined value means the best classification results for the dataset. 2) The only underlined value means the best classification results for the dataset compared with a specific technique.

6.1. Word Normalisation

Based on the results in Table 6, we tended to focus more on the effectiveness of using different features for the text in Arabic. The attached pronoun in Arabic text is one of the biggest challenges in the Arabic NLP because of changing the form of a single word into multiple forms. In this paper, we investigate this more to better prepare Arabic text features to normalise the text, which was detailed in Subsection 4.1. The approach was either by splitting the attached pronouns from the core of the word or trimming the pronouns from the word itself. We used three different tools for preprocessing the Arabic words; MADAMIRA (Pasha et al., 2014), Farasa (Abdelali et al., 2016) and Stanford (Manning et al., 2014). In these techniques, the tokens were represented by vectors that were generated automatically from the input dataset. The results of using the proposed deep learning classification model for all the datasets with all the Word Normalisation techniques are shown in Table 7. The highest results for each dataset are emboldened. Table 7 indicates that for each dataset we can obtain better results using different techniques. For the Main-AHS dataset, the Farasa Lemmatisation tool has the best sentiment classification results. While the best accuracy result for the Sub-AHS dataset is obtained using the Stanford Segmentation technique. For the Ar-Twitter dataset, the highest result is using the Madamira Stem approach. Finally, all the segmentation techniques using the three tools namely, Madamira, Farasa, and Stanford, show the top results for the ASTD dataset. The incompatibility in the techniques is due to the nature of the text in each dataset.

The variations in the results using the six different techniques are very small for both the Sub-AHS and Ar-Twitter datasets, which are 0.0173 and 0.0152, respectively. Also, for the ASTD dataset, we see a small variation. In contrast, there is a clear variation in the accuracy results for the Main-AHS, which reaches 0.0320. Fig. 3 shows the accuracy results over 50 epochs for all four datasets using all six approaches for Word Normalisation. Also, it clarifies more the variations of the accuracy results based on different Word Normalisation techniques. For the Main-AHS dataset, the line chart of Farasa Lemmatisation always clearly has a higher level after the fifth epoch compared with other techniques. For the Sub-AHS, the accuracy line charts have small gaps between 0.9350 and 0.9650. Stanford Segmentation techniques generally show the top results in accuracy after the eighth epoch compared with other Word Normalisation approaches. For the Ar-Twitter dataset, the accuracy variance in the line charts is very limited, from 0.7730 to 0.8890. Thus, the techniques have very similar classification results, and the Madamira Stem technique has the best result overall compared with other techniques. For the ASTD dataset, most of the accuracy lines charts zigzag until the thirtieth epoch when they become steadier. Madamira Segmentation, Farasa Segmentation and Stanford Segmentation techniques generally have the best classification performance. Table 7 shows that the Farasa Segmentation technique generally proves to be the best approach for the used four datasets. Also, the example in Table 4 clarifies the ability of this approach in splitting the core of the word from the prefixes and suffixes.

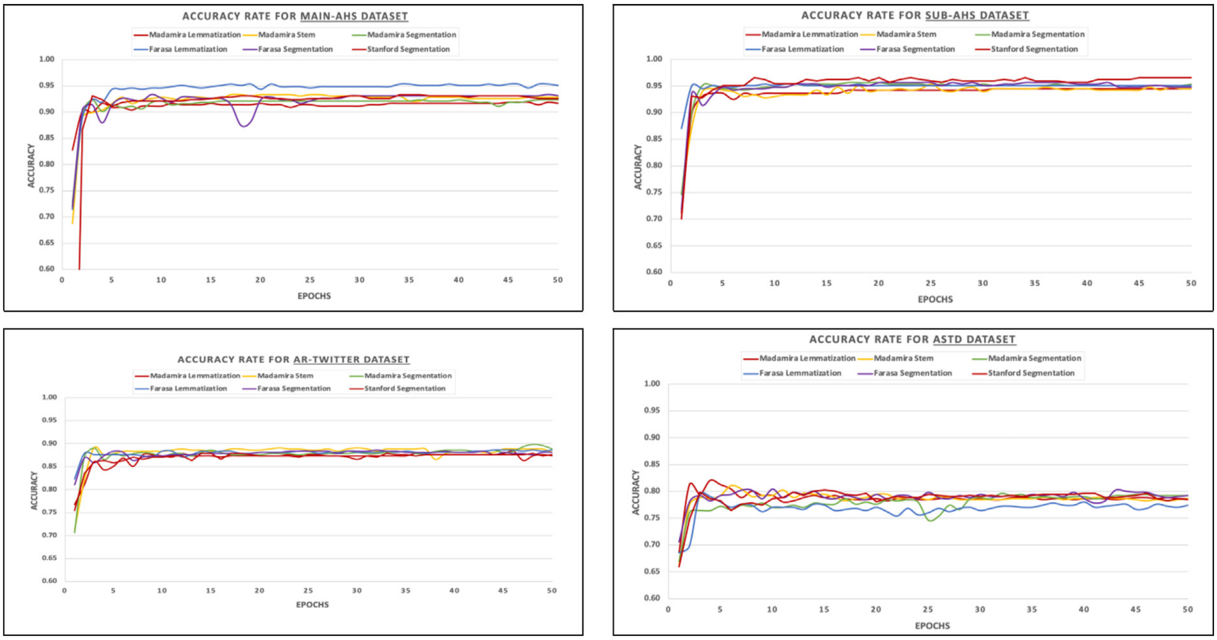

Fig. 3. Accuracy for the proposed model on the test set for all the datasets using Word Normalisation techniques.

Table 7
Accuracy comparison of the proposed method with different word normalisation techniques on different datasets.

	Main-AHS	Sub-AHS	Ar-Twitter	ASTD
Madamira Lemmatisation	0.9163	0.9452	0.8734	0.7863
Madamira Stem	0.9286	0.9452	**0.8886**	0.7883
Madamira Segmentation	0.9212	0.9510	0.8785	**0.7923**
Farasa Lemmatisation	**0.9483**	0.9510	0.8810	0.7702
Farasa Segmentation	0.9310	0.9568	0.8810	**0.7923**
Stanford Segmentation	0.9310	**0.9625**	0.8759	**0.7923**

1) The emboldened and underlined value means the best classification results for the dataset. 2) The only underlined value means the best classification results for the dataset compared with a specific technique.

6.2. Word Embedding

We used different pre-trained word embedding techniques for the sentiment classification to represent the text features. The word embedding techniques were Word2Vec (CBOW and SG), Glove and fastText, both (CBOW and SG). Each model has three different dimensions, which are 100, 200 and 300, to determine the role of using different lengths of vectors, and this was mentioned in Subsection 4.2 This approach was applied to all the datasets using the word levels, which is on the original datasets without any text pre-processing. Table 8 shows the accuracy of the sentiment classification results for the four datasets using different word embedding techniques with different dimensions.

It is obvious from Table 8 and Figs. 4–6 that the fastText CBOW approach has the smallest accuracy results for all the four datasets with all the different dimensions compared with other approaches.

Table 8
Accuracy comparison of the proposed method with different word embedding techniques with three dimensions.

	Dim	Main-AHS	Sub-AHS	Ar-Twitter	ASTD
Word2Vec SG	100	0.9111	0.9408	0.8709	0.7657
	200	**0.9358**	0.9509	0.8760	**0.8162**
	300	**0.9358**	**0.9668**	**0.8848**	0.7859
Word2Vec CBOW	100	0.9160	0.9495	0.8545	0.7879
	200	0.9297	0.9610	0.8646	0.7940
	300	**0.9358**	0.9653	0.8760	0.8051
GloVe	100	0.9284	0.9451	0.8519	0.7899
	200	0.9235	0.9408	0.8722	0.7859
	300	0.9309	0.9480	0.8646	0.7950
fastText SG	100	0.9235	0.9466	0.8760	0.8000
	200	0.9235	0.9610	0.8633	0.8051
	300	0.9210	0.9610	0.8760	0.8101
fastText CBOW	100	0.8926	0.9321	0.8291	0.7434
	200	0.8963	0.9249	0.8355	0.7384
	300	0.8840	0.9379	0.8355	0.7647

Table 9
Accuracy comparisons of the best results of the proposed model with other models on the same datasets.

Models	Techniques		Datasets			
			Main-AHS	Sub-AHS	Ar-Twitter	ASTD
The proposed model	Madamira Segmentation					0.7923
	Farasa Lemmatisation		**0.9483**			
	Farasa Segmentation					0.7923
	Stanford Segmentation			0.9625		0.7923
	Madamira Stem				**0.8886**	
	Word level		0.9335	0.9539	0.8861	0.7823
	Word2VecSG	200	0.9358			**0.8162**
		300	0.9358	0.9668	0.8848	0.7859
	Word2VecCBOW	300	0.9358			
The model in (Alayba et al., 2018)	Ch5gram- Level			0.9568		0.7762
	Word-Level		0.9424		0.8810	
Other models	The model in (Al Omari et al., 2019)		0.881	**0.968**	0.842	0.7918
	The model in (Nabil et al., 2015)				0.8501	0.7907
	The model in (Abdulla et al., 2013)				0.872	

1) The emboldened and underlined value means the best classification results for the dataset. 2) The only underlined value means the best classification results for the dataset compared with a specific technique.

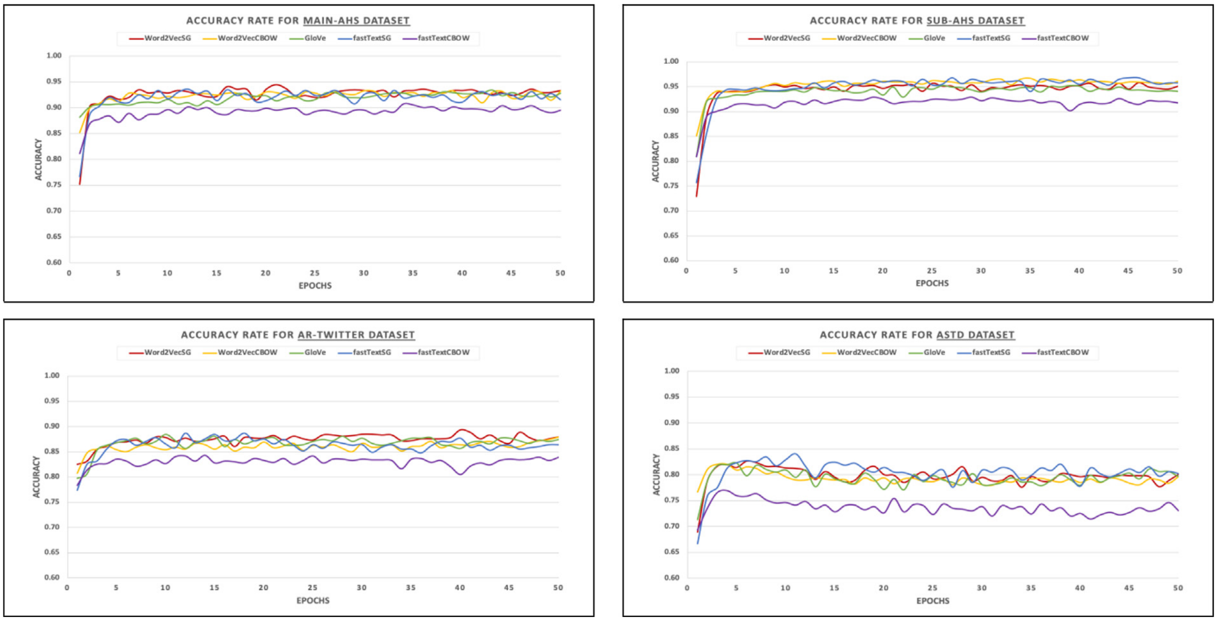

Fig. 4. Accuracy for the proposed model on the test set for all the datasets using Word Embedding techniques with 100 dimensions.

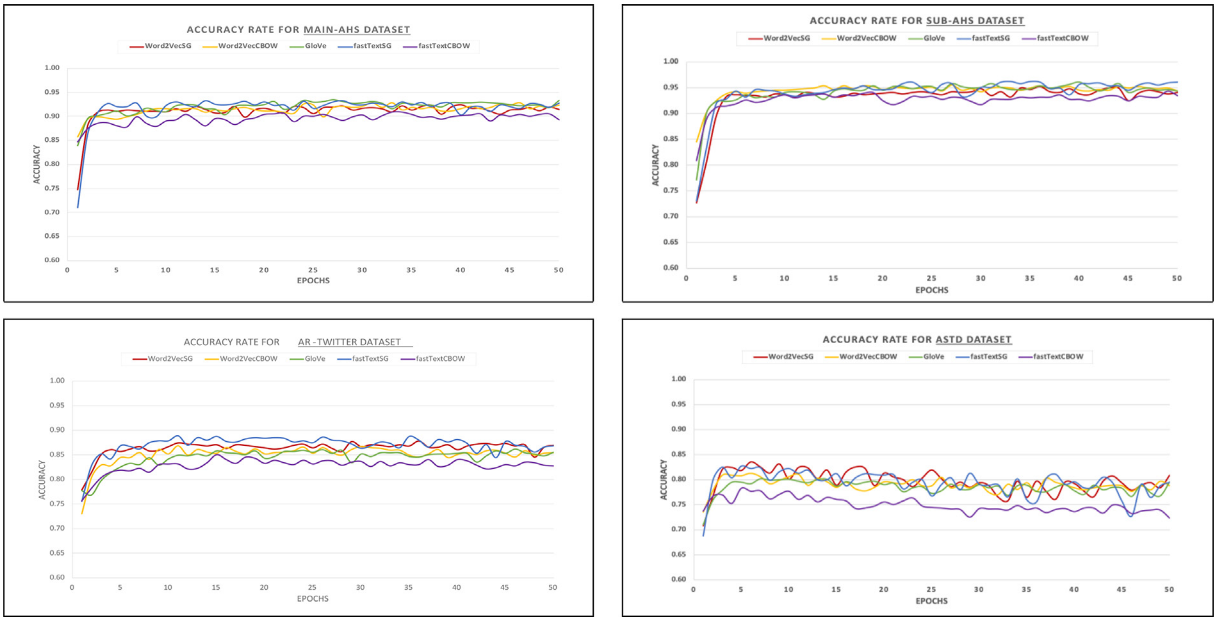

Fig. 5. Accuracy for the proposed model on the test set for all the datasets using Word Embedding techniques with 200 dimensions.

Also, most of the results in Table 8 show that increasing the length of the vectors for pre-trained word embedding techniques leads to better classification results. Moreover, Table 8 illustrates the strength of using Word2Vec (SG) techniques as a pre-trained word embedding on the four datasets for sentiment classification purposes. It has superior accuracy in its results using 300 dimensions for all the datasets except the ASTD dataset; while it shows the best performance using 200 dimensions for the ASTD dataset. For the Main-AHS dataset, the Word2Vec (CBOW) technique shares the same performance of text classification using 300 dimensions.

6.3. Comparing the proposed classifier results with other approaches

In order to examine the efficiency of our proposed Arabic text classification approach, we compared the best classification results of our developed CNN-LSTM model with other models. Table 9 summarises the best Arabic text classification results using the proposed models for word normalization and word embedding feature preparation techniques. Additionally, it compares them with the previous CNN-LSTM model and other text classification models for the same datasets. The enhanced CNN-LSTM model achieves the highest results for three datasets,

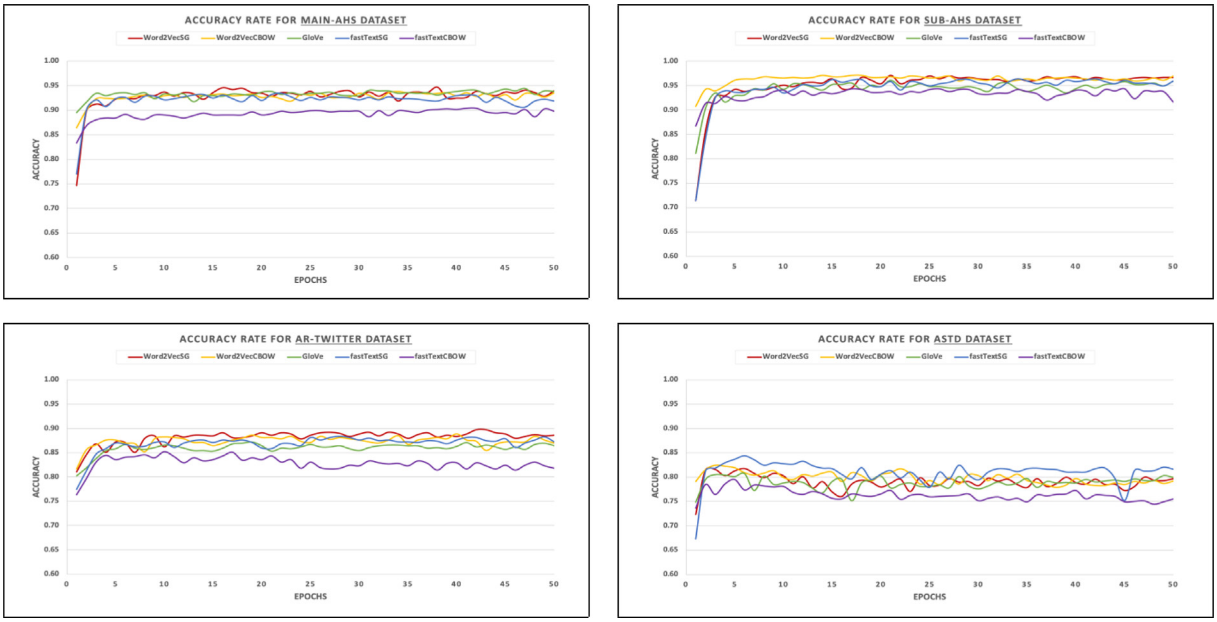

Fig. 6. Accuracy for the proposed model on the test set for all the datasets using Word Embedding techniques with 300 dimensions.

which are the Main-AHS, Ar-Twitter, and ASTD. For the Main-AHS dataset, the highest result is 0.9483 and it was obtained using the Word Normalisation approach (Farasa Lemmatisation). For the Ar-Twitter dataset, the best acquired result is 0.8886 and it was obtained using the Word Normalisation approach (Madamira Stem). For the ASTD dataset, the top result is 0.8162 and it was reached using the Word Embedding approach (Word2VecSG, 200 dimensions). However, the model in (Al Omari et al., 2019) has better result for the Sub-AHS dataset, which was 0.9680 compared to our best result, which was 0.9668.

7. Conclusions and future work

This paper considered the effectiveness of different methods of preparing Arabic text in order to better represent the text features. The richness in the Arabic morphology was investigated by reducing the forms of Arabic words using word normalization techniques. Moreover, the semantics of Arabic words were examined using different word embedding techniques with different dimensions. Also, this research studied the effectiveness of combining CNN and LSTM networks and deleting the Max-pooling layer in the CNN. We proposed a novel method by integrating Arabic word normalisation tools with an effective sentiment analysis classification model that better represents the features. Also, we build multiple Arabic word embedding models using the same corpora to measure the leverage of a variety of Arabic word representations on the sentiment classification. We proposed an Arabic sentiment classification model that has shown state-of-the-art results using different techniques for representing the features of Arabic text.

Future work needs to consider more complex architectures for CNN and LSTM networks, or other alternative deep learning algorithms. Also, larger sentiment analysis datasets need to be used with more complex neural network models. Furthermore, different word embedding techniques will be used and investigated, such as Elmo (Peters et al., 2018). Using the BERT model and transformers techniques for language representation (Devlin et al., 2019) have shown very promising results in NLP tasks so far, and it will also be one future line of investigations for us.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdelali, A., Darwish, K., Durrani, N., Mubarak, H., 2016. Farasa: A Fast and Furious Segmenter for Arabic. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 11–16.

N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based," in *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Dec. 2013, vol. 6, no. 12, pp. 1–6.

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*, pp. 350–357.

I. Abu El-khair, "1.5 billion words Arabic Corpus," 2016, [Online]. Available: http://arxiv.org/abs/1611.04033.

Abu Kwaik, K., Saad, M., Chatzikyriakidis, S., Dobnik, S., 2019,. "LSTM-CNN Deep Learning Model for Sentiment Analysis of Dialectal Arabic", in *Arabic Language Processing: From Theory to Practice*. ICALP 2019, 108–121.

Adnan, K., Akbar, R., 2019. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. Int. J. Eng. Bus. Manag. 11.

T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep Audio-Visual Speech Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–13, 2018.

Agarwal, B., Mittal, N., 2016. Machine Learning Approach for Sentiment Analysis, 21–45.

Ahuja, R., Chug, A., Kohli, S., Gupta, S., Ahuja, P., 2019. The Impact of Features Extraction on the Sentiment Analysis. Procedia Comput. Sci. 152, 341–348.

M. Al Omari, M. Al-Hajj, A. Sabra, and N. Hammami, "Hybrid CNNs-LSTM Deep Analyzer for Arabic Opinion Mining," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Oct. 2019, pp. 364–368.

Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El Hajj, W., Bashir Shaban, K., 2015. Deep Learning Models for Sentiment Analysis in Arabic. In: *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pp. 9–17.

A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving Sentiment Analysis in Arabic Using Word Representation," in *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, Mar. 2018, pp. 13–18.

Alayba, A., Palade, V., England, M., Iqbal, R., 2017. Arabic Language Sentiment Analysis on Health Services. In: *1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pp. 114–118.

Alayba, A.M., Palade, V., England, M., Iqbal, R., 2018. A Combined CNN and LSTM Model for Arabic Sentiment Analysis. Cross Domain - Machine Learning and Knowledge Extraction CD-MAKE 2018, 179–191.

Al-Azani, S., El-Alfy, E.-S.-M., 2017. Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs. Neural Information Processing. ICONIP 2017, 491–500.

Almuqren, L., Cristea, A.I., 2016. Framework for Sentiment Analysis of Arabic Text. In: *Proceedings of the 27th ACM Conference on Hypertext and Social Media - HT '16*, pp. 315–317.

Al-Rowaily, K., Abulaish, M., Al-Hasan Haldar, N., Al-Rubaian, M., 2015. BiSAL – A bilingual sentiment analysis lexicon to analyze Dark Web forums for cyber security. Digit. Investig., Sep. 14, 53–62.

Al-Sallab, A., Baly, R., Hajj, H., Shaban, K.B., El-Hajj, W., Badaro, G., Sep. 2017. AROMA: A Recursive Deep Learning Model for Opinion Mining in Arabic as a Low Resource Language. ACM Trans Asian Low-Resource Lang. Inf. Process. 16 (4), 1–20.

Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., Gupta, B., Jul. 2018. Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. J. Comput. Sci. 27, 386–393.

M. Althobaiti, U. Kruschwitz, and M. Poesio, "AraNLP: a Java-based Library for the Processing of Arabic Text," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014, pp. 4134–4138, [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/621_Paper.pdf.

Altowayan, A.A., Elnagar, A., 2017. Improving Arabic sentiment analysis with sentiment-specific embeddings. In: *2017 IEEE International Conference on Big Data (Big Data)*, Dec, pp. 4314–4320.

Aqib, M., Mehmood, R., Albeshri, A., Alzahrani, A., 2018. "Disaster Management in Smart Cities by Forecasting Traffic Plan Using Deep Learning and GPUs", in *Smart Societies*. Technologies and Applications, Infrastructure, pp. 139–154.

Ashi, M.M., Siddiqui, M.A., Nadeem, F., 2018,. Pre-trained Word Embeddings for Arabic Aspect-Based Sentiment Analysis of Airline Tweets. In: *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, pp. 241–251.

Baly, R., Hajj, H., Habash, N., Shaban, K.B., El-Hajj, W., Sep. 2017. A Sentiment Treebank and Morphologically Enriched Recursive Deep Models for Effective Sentiment Analysis in Arabic. ACM Trans Asian Low-Resource Lang. Inf. Process. 16 (4), 1–21.

Baytas, I.M., Xiao, C., Zhang, X., Wang, F., Jain, A.K., Zhou, J., 2017. Patient Subtyping via Time-Aware LSTM Networks. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 65–74.

Bian, J., Gao, B., Liu, T.-Y., 2014. Knowledge-Powered Deep Learning for Word Embedding. Machine Learning and Knowledge Discovery in Databases 5211, 132–148.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., Dec. 2017. Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguist. 5, 135–146.

J. Chen, N. Tandon, and G. de Melo, "Neural Word Representations from Large-Scale Commonsense Knowledge," *2015 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, pp. 225–228, Dec. 2015.

K. Cheng, J. Li, J. Tang, and H. Liu, "Unsupervised Sentiment Analysis with Signed Social Networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Feb. 2017, vol. 31, no. 1 SE-Main Track: NLP and Text Mining, [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11008.

F. Chollet and others, "Keras," 2020. https://keras.io/ (accessed May 02, 2020).

A. Dahou, S. Xiong, J. Zhou, M. Haddoud, and P. Duan, "Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification.," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics:*, 2016, pp. 2418–2427, [Online]. Available: http://www.aclweb.org/old_anthology/C/C16/C16-1228.pdf.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun. 2019, pp. 4171–4186.

Dey, L., Chakraborty, S., Biswas, A., Bose, B., Tiwari, S., Jul. 2016. Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier. Int. J. Inf. Eng. Electron. Bus. 8 (4), 54–62.

dos Santos, C., Gatti, M., 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78.

D. M. El-Din, "Enhancement Bag-of-Words Model for Solving the Challenges of Sentiment Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, 2016.

T. Elghazaly, A. Mahmoud, and H. A. Hefny, "Political Sentiment Analysis Using Twitter Data," in *Proceedings of the International Conference on Internet of things and Cloud Computing - ICC '16*, 2016, pp. 1–5.

El-Kilany, A., Azzam, A., El-Beltagy, S.R., 2018. Using Deep Neural Networks for Extracting Sentiment Targets in Arabic Tweets. Intell. Nat. Lang. Process. Trends Appl., 3–15

P. Etoori, M. Chinnakotla, and R. Mamidi, "Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning," in *Proceedings of ACL 2018, Student Research Workshop*, 2018, pp. 146–152.

Fang, X., Zhan, J., Dec. 2015. Sentiment Analysis Using Product Review Data. J. Big Data 2 (1), 5.

Python Software Foundation, "Python," 2020. https://www.python.org/ (accessed Feb. 20, 2020).

Python Software Foundation, "ElementTree," 2020. https://docs.python.org/2/library/xml.etree.elementtree.html (accessed Apr. 09, 2020).

Han, J., Moraga, C., 1995. The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning. From Nat. Artif Neural Comput., 195–201

Harris, Z., 1954. Distributional Structure. Word 10 (23), 146–162.

Heikal, M., Torki, M., El-Makky, N., 2018. Sentiment Analysis of Arabic Tweets using Deep Learning. Procedia Comput. Sci. 142, 114–122.

J. Hwang and Y. Zhou, "Image Colorization with Deep Convolutional Neural Networks," 2016. [Online]. Available: http://cs231n.stanford.edu/reports/2016/pdfs/219_Report.pdf.

M. M. Itani, R. N. Zantout, L. Hamandi, and I. Elkabani, "Classifying sentiment in arabic social networks: Naïve search versus Naïve bayes," in *2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, Dec. 2012, pp. 192–197.

Jagdale, R.S., Shirsat, V.S., Deshmukh, S.N., 2019. Sentiment Analysis on Product Reviews Using Machine Learning Techniques, 639–647.

Kaibi, I., Nfaoui, E.H., Satori, H., 2020. Sentiment analysis approach based on combination of word embedding techniques. In: *Embedded Systems and Artificial Intelligence*, pp. 805–813.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. (Volume 1 Long Pap.*, pp. 655–665, 2014.

Khan, F.H., Qamar, U., Bashir, S., Jun. 2017. A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet. Knowl. Inf. Syst. 51 (3), 851–872.

Khoo, C.S., Johnkhan, S.B., Aug. 2018. Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. J. Inf. Sci. 44 (4), 491–511.

Kim, Y., Sep. 2014. Convolutional Neural Networks for Distant Speech Recognition. IEEE Signal Process Lett. 21 (9), 1120–1124.

Liu, B., 2015. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, Cambridge.

Liu, D., Cheng, B., Wang, Z., Zhang, H., Huang, T.S., Sep. 2019. Enhance visual recognition under adverse conditions via deep networks. IEEE Trans. Image Process. 28 (9), 4401–4412.

Mahata, S.K., Das, D., Bandyopadhyay, S., Jul. 2019. MTIL2017: machine translation using recurrent neural network on statistical machine translation. J. Intell. Syst. 28 (3), 447–453.

Manning, C.D., Raghavan, P., Schutze, H., 2008. Introduction to Information Retrieval. Cambridge University Press, Cambridge.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D., 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60.

K. Merchant and Y. Pande, "NLP Based Latent Semantic Analysis for Legal Text Summarization," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2018, pp. 1803–1807.

T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," pp. 1–12, Jan. 2013, [Online]. Available: http://arxiv.org/abs/1301.3781.

D.J. Mona Diab, Kadri Hacioglu, "Automated Methods for Processing Arabic Text: from Tokenization to Base Phrase Chunking," in *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, 2007.

Monroe, W., Green, S., Manning, C.D., 2014. Word Segmentation of Informal Arabic with Domain Adaptation. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 206–211.

M. Nabil, M. Aly, and A. Atiya, "ASTD: Arabic Sentiment Tweets Dataset," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, no. September, pp. 2515–2519, [Online]. Available: http://aclweb.org/anthology/D15-1299.

Nair, G.E., Vinod and Hinton,, 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In: *The 27th International Conference on International Conference on Machine Learning ICML*, pp. 807–814.

M. S. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Jul. 2013, pp. 1–5.

M. Nickel and D. Kiela, "Poincaré Embeddings for Learning Hierarchical Representations," in *Neural Information Processing Systems (Nips)*, 2017, pp. 6338–6347, [Online]. Available: http://arxiv.org/abs/1705.08039.

Nizar Habash, N.T., Roth, R., Rambow, O., Eskander, R., 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 426–432.

A. Pasha, M. Al-Badrashiny, M. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, R. Roth, "MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 14)*, 2014, pp. 1094–1101.

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)*, pp. 1532–1543.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep Contextualized Word Representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237.

Poliak, A., Belinkov, Y., Glass, J., Van Durme, B., 2018. On the Evaluation of Semantic Phenomena in Neural Machine Translation Using Natural Language Inference. In: *Proceedings of the 2018 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 513–523.

A. Rahman and M. S. Hossen, "Sentiment Analysis on Movie Review Data Using Machine Learning Approach," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sep. 2019, pp. 1–4.

Rao, Q., Frtunikj, J., 2018. Deep Learning for Self-Driving Cars: Chances and Challenges. In: *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems - SEFAIS '18*, pp. 35–38.

Ribeiro, B., Oliveira, G., Laranjeira, A., Arrais, J.P., 2017. Deep learning in digital marketing: brand detection and emotion recognition. Int. J. Mach. Intell. Sens. Signal Process. 2 (1), 32.

Ribeiro, M.T., Singh, S., Guestrin, C., 2018. Semantically Equivalent Adversarial Rules for Debugging NLP Models. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865.

Rizun, N., Taranenko, Y., Waloszek, W., Dec. 2018. Improving the Accuracy in Sentiment Classification in the Light of Modelling the Latent Semantic Relations. Information 9 (12), 307.

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., Beling, P., 2018. Deep Learning Detecting Fraud in Credit Card Transactions. In: *2018 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 129–134.

Sadr, H., Pedram, M.M., Teshnehlab, M., Dec. 2019. A robust sentiment analysis method based on sequential combination of convolutional and recursive neural networks. Neural Process. Lett. 50 (3), 2745–2761.

B. Shin, T. Lee, and J. D. Choi, "Lexicon Integrated CNN Models with Attention for Sentiment Analysis," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Sep. 2017, pp. 149–158, [Online]. Available: http://aclweb.org/anthology/W17-5220.

R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions," in Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 151–161, [Online]. Available: https://www.aclweb.org/anthology/D11-1014.

R. Socher A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642, [Online]. Available: https://www.aclweb.org/anthology/D13-1170.

Soliman, A.B., Eissa, K., El-Beltagy, S.R., 2017. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. Procedia Comput. Sci. 117, 256–265.

Van, V.D., Thai, T., Nghiem, M.-Q., 2017,. Combining convolution and recursive neural networks for sentiment analysis. Proc. Eighth Int. Symp. Inf. Commun. Technol. - SoICT, 151–158.

Vilares, D., Alonso, M.A., Gómez-Rodríguez, C., May 2017. Supervised sentiment analysis in multilingual environments. Inf. Process. Manag. 53 (3), 595–607.

J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model," Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 2 Short Pap., pp. 225–230, 2016.

Wang, X., Jiang, W., Luo, Z., 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2428–2437.

Wang, X., Liu, Y., Sun, C., Wang, B., Wang, X., 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. 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)*, pp. 1343–1353.

Xu, A., Liu, Z., Guo, Y., Sinha, V., Akkiraju, R., May 2017. A new chatbot for customer service on social media. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 3506–3510.

Xu, G., Meng, Y., Qiu, X., Yu, Z., Wu, X., 2019. Sentiment Analysis of Comment Texts Based on BiLSTM. IEEE Access 7, 51522–51532.

Yang, L., Li, Y., Wang, J., Sherratt, R.S., 2020. Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. IEEE Access 8, 23522–23530.

Yu, L.-C., Wu, J.-L., Chang, P.-C., Chu, H.-S., Mar. 2013. Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news. Knowledge-Based Syst. 41, 89–97.