Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-07993-4

Opinion mining in Persian language using a hybrid feature extraction approach based on convolutional neural network

Shima Zobeidi [1] · Marjan Naderan [1] · Seyyed Enayatallah Alavi [1]

Received: 2 November 2018 / Revised: 1 June 2019 / Accepted: 10 July 2019
Published online: 02 August 2019

Abstract
Nowadays, huge amounts of text data are generated due to the increase of communications, over various web sites and applications. Evaluation and extraction of information from these data is an important task, one way of which is named opinion mining. The purpose of this paper is sentiment analysis of users' opinions about various products. The proposed system classifies opinions at the sentence level based on emotions into two and multiple classes by deep learning methods. To this end, three main phases are taken: the first step contains sentences preparation for the input matrix which itself is accomplished in two levels: word-level and character-level. In word-level, each word in each sentence is given to the word2vec algorithm. In character-level, for each character in each sentence, the proposed method computes a numerical vector and creates a matrix. Next, the feature extraction part is executed which includes a Convolutional Neural Network (CNN). The generated matrices in the previous levels for each sentence are given to the CNN for embedding each sentence and therefore, utilizing both word2vec and CNN for extracting features. In the final step, the generated vectors are given to the Bidirectional Long Short Term Memory (Bi-LSTM) network for sentiment classification, not used in any of the previous methods. The performance of the proposed algorithm has been investigated on the Digikala Persian dataset on mobile and digital cameras. Results show that the proposed algorithm reaches an accuracy of 95% for two classes and 92% for multi-class classification which is comparable with previous algorithms.

Keywords Deep learning · Text mining · Opinion mining · Convolutional neural network · Bi-LSTM · Word2vec · Character-level · Sentiment analysis

✉ Marjan Naderan
m.naderan@scu.ac.ir

[1] Department of Computer Engineering, Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran

1 Introduction

Nowadays, with the development of communication tools and the creation of social networks such as Instagram, Facebook, Twitter, and Telegram, the Internet plays a significant role in people's lives. Online shopping sites have been popular, on many of which people can express the disadvantages and advantages of the products they have purchased. People consult with others and read online comments about products before buying them.

User comments are evaluated using the distribution of polarity ratings (criteria), which refers to identifying positive, negative, and neutral aspect of subjective sentences [45]. These evaluations are accomplished by sentiment analysis techniques. Sentiment analysis has attracted huge attention and has various applications in different fields such as recommender systems [24], market forecasting [25], social network analysis [22], medical domain [17], and the prediction of political topics [36]. One of the purposes of sentiment analysis is to distinguish the polarity of the opinions expressed in a text. In fact, one well-known application of sentiment analysis is opinion mining. Opinion mining uses Natural Language Processing (NLP) and data mining techniques to extract useful information from a large amount of comments written by customers [29]. Opinion mining is done in three levels: document-level, sentence-level and aspect-level [40].

In the document-level [46] opinion mining, the goal is to identify the positive or negative polarity of the entire document. In sentence-level [26], [47], in the first step, sentences are identified; then, subjective sentences are determined, and in the final step, the positive or negative polarity of the sentences is determined. In aspect-level [38] opinion mining, the purpose is to identify the various aspects of different entities and to determine the positivity or negativity for each aspect of the different entities. For example, the sentence "the pictures coming out of this cellphone are amazing and it has a good battery life" talks about two different aspects of the cellphone, which are the quality of the camera and battery.

There are numerous challenges in the field of text mining, especially in opinion mining. One of them is the language of the opinions. For instance, many researches have been conducted in the field of opinion mining on English [27], Chinese [27], and Arabic [9] languages but few researches are available for the Persian language. Persian language is common in Iran, Afghanistan, and a part of Tajikistan and it has specific challenges such as different dialects, foreign words, different structure, conceptual diversity, etc. [7].

In this paper, opinion mining is performed in sentence-level on a well-known Persian dataset. The sentence-level approach is chosen due to the fact that working in document-level increases the dimensions of the matrices and hence, reduces the performance and accuracy of the whole approach. In addition, using the techniques of deep learning, the polarity of users' comments in Persian are determined. Deep learning techniques are powerful algorithms that have a significant role in computer vision [12, 44], speech recognition [37, 39], and NLP [34, 50]. Deep neural networks are types of deep learning methods such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Recursive Neural Network. For dependent and sequential data, RNNs have a high ability in learning [11]. In the RNNs, the patterns of input are captured in the hidden layers, which increases the efficiency for the sequential data. The simplest RNN [11, 14, 21, 31] cannot remember information for a long period of time while LSTM is designed for long-term dependency, and is used for different types of data such as: audio, video, and text. Also, the LSTM network is used for the purpose of classification and prediction [28, 35].

The input form for these networks is important. In this work, the input is given in two ways to the network: word-based and character-based. Each of them has its own advantages and disadvantages, and should be used according to the circumstances. The main advantage of the character-based input over the word-based one is that it has a really small vocabulary. In practice, this means that character-based input requires less memory and has faster inference than the word-based method. Another advantage of character-based input is that it does not require tokenization as a preprocessing step (the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens). But most of the time, word-based methods yield higher accuracy values and have a lower computational cost.

In this work, CNN is used for feature extraction. In the word-level approach, first each sentence is separated to its words and next, the words of each sentence are transformed to the vector space by the word2vec algorithm [32]. Therefore, to represent a sentence, a matrix is calculated in which the rows of the matrix are equal to the number of words and the columns are variably calculated according to the word2vec algorithm. This matrix is given as input to the CNN. In the character-based approach, first each sentence is separated to its characters and next the characters of each sentence are represented by a vector using the one-hot method. All characters of a sentence constitute a matrix, which is given as input to the CNN. In fact, we use word- and character-level features separately and each of them results in a different input matrix for the feature extraction module. Finally, the output of the feature extraction step is given to Bi-LSTM for sentiment classification. In addition, the dataset collected from the Digikala website [13] is used which contains a total of 151229 customer reviews about cellphones and digital cameras. The Digikala web site is one of the largest and most well-known websites in Persian.

In short, the innovations of this work can be summarized as:

- Using both word- and character-level sentiment analysis using the traditional word2vec method and the new CNN approach;
- Using the Bi-LSTM network for classification of users' comments, which have not been applied in any of the previous works;
- Investigating both two-class and multi-class classification and analyzing the performance of each one. In most of the previous works, two-class classification is investigated and few contain multi-class ones.
- Examining the Persian dataset of the well-known site Digikala on considering comments about cellphones and digital cameras.

Overall, the novelty in general (which does not rely on the type of language) is using both Word2vec and CNN in the feature extraction phase and using Bi-LSTM for classification. The novelty of this work, specifically in Persian, is using the Digikala dataset which has a huge database of users' comments written in formal and informal expressions, which we stated in the preprocessing phase. In fact, the preprocessing phase applied to Digikala dataset, raised innovations specific in Persian language which were not mentioned in any of the works on Persian [2, 5, 8, 41].

The rest of the paper is organized as follows: In Section 2, some of the related work in the field of opinion mining, both in Persian and other languages, are presented. In Section 3, the background knowledge of the structures and components of the proposed method is expressed. In Section 4, the proposed method, with details of the steps is introduced. Section 5 describes

the dataset and reports the experimental results. Finally, Section 6 concludes the paper with some directions for future works.

2 Related work

Representation of words and sentences in vector space is the most important component in opinion mining.

Socher et al. [42] proposed a method that is based on semi-supervised recursive auto-encoders to predict the emotional labels. In this method, variable-sized phrases are converted to vectors using hierarchical structures. At the end, a Softmax function is applied on the vector representation of phrases to predict labels. The dataset they have used consists of users personal stories annotated with multiple labels.

In [43], the main idea is the combination of matrix-vector representations with a recursive neural network (MV-RNN). This model is built based on a binary parse tree. MV-RNN can learn the meaning of operators in propositional logic and natural language.

As s popular technique, Word2vec is used as a method of representing words in the vector space in a variety of ways. For instance, in [15], opinions are classified using word2vec in the document level. Text documents must be converted to numeric vectors for classification. In this work, text documents are converted to numeric vectors using three methods: word2vec, bag of words (BOW) and BOW+word2vec. Then, the Logistic Regression (LR) algorithm classifies the numeric data. These three methods have been simulated and the results show that BOW+word2vec approach yields a higher accuracy.

In [1], feature extraction is accomplished using a standard sentiment lexical dictionary and word2vec. In general, the feature extraction structure consists of three main components: (1) Learning Word Representation based on word2vec, (2) clustering of terms in vocabulary based on opinion words, and (3) construction of features matrix based on cluster centroids for classification. For evaluation, the Movie Review Dataset (ACLIMDB) is used, which is available online. Two classification algorithms are used, namely Logistic Regression and Support Vector Machine (SVM). The proposed feature extraction method is compared with Word2vec, Doc2vec [23] and bag-of-words methods and the results show that the proposed method has a higher accuracy.

As an applicable approach, opinion mining is used in various fields such as medical, economic, and political domain. Gopalakrishnan et al. in [17] try to examine opinions of patients about the qualities and prices of two different types of important drugs. For this purpose, two neural networks are used, namely Probabilistic Neural Network (PNN) and Radial Basis Function Neural network (RBFN) that are compared with SVM. Also the Term Frequency-Inverse Document Frequency (TF-IDF) technique is used for feature extraction. Finally, the evaluation shows that RBFN has a higher accuracy than PNN and SVM.

In [10], a model is proposed that incorporates a combination of dependency parser, sub-tree mining, and deep learning for sentiment classification. In this model, deep neural networks, namely the LSTM and Gated Recurrent Neural Network (GRNN) were used. The proposed model first encodes the relation among words using the dependency tree and finds the best sub-tree in the outlier detection phase. Finally, sentences are classified using LSTM+GRNN. To evaluate performance, the proposed model has been applied on multi-domain datasets.

Numerous deep architectures have been proposed in many works, e.g., auto-encoder, recursive neural network, and LSTM for example in [10, 17, 42], where the CNN is on the

center of attention. For instance, in [16], a combination model called Boosted Convolution Neural Network for sentiment analysis was proposed. In a part of CNN, different filters were employed with different window sizes to scan the input sentences. Using the AdaBoost method, classification results with different classifiers weights are aggregated and the final polarity decision about the positive or negative class is achieved. Two common datasets are used for this model that contain movie reviews in positive/negative classes. The proposed model was compared with other methods and the results show that the proposed model outperforms previous methods.

In [30], a hybrid system was proposed based on machine learning and rule-based approaches to create a de-identification system for the 2014 i2b2 NLP challenge. The system contains two machine learning-based classifiers and a rule-based classifier. In this system, instances are first identified by two (token-level and character-level) Conditional Random Fields (CRFs) and a rule-based classifier, and next they are merged by some rules. The character-level CRFs outperform the token-level CRFs. When all the three classifiers are used, the system is further improved with the best micro F-score of 91.24%.

An et al. in [3] proposed a method for sentiment analysis on short Chinese texts with a character-level view. Segmenting a sentence into words is a much harder process in tonal languages, such as Chinese and Thai, than the others, such as English. Thus, in this work, numerous algorithms are implemented only based on character level features to avoid this problem. These algorithms include character-level NB, character-level SVM and character-level CNN. Furthermore, to test the effect of character-level features, they are compared with word-level NB and word-level SVM. Results show that the character-level models preserve more information than the word-level ones and thus yield a better accuracy.

Finally, to introduce studies on Persian language, the first paper addressing opinion mining in Persian language was published in 2012 by Shams et al. [41]. They proposed an unsupervised LDA-based method and applied their method on three manually-created small review datasets. In another study, Bagheri and Saraee in [5] proposed an approach on opinion mining in Persian language that is also an ML-based method. The first lexicon-based method for opinion mining in Persian Language was proposed by Basiri etal. In [8], in which a new framework was proposed. In [2], another lexicon-based opinion mining method was proposed and an improvement over other lexicon-based methods in terms of accuracy was reported.

At the end of this section and as a conclusion, the most important methods are presented in Table 1.

3 Materials and methods

In this section, we introduce the models of word2vec, convolutional neural network and long short-term memory as basics for learning the sentence embedding vectors and classification.

3.1 Word2vec

Word2vec algorithm is one of the natural language processing (NLP) tools that was proposed by Tomas Mikolov and his colleagues in 2013 [32]. This algorithm is a word embedding method that maps words into vector spaces. Embedding of words is according to the meaning of words and their semantic relation with the adjacent words in text. It is suitable for semantic analysis and yields high accuracy values. Word2vec is provided by combining two main

Table 1 Comparison of the most important methods in the field of sentiment analysis

Reference	Feature selection method	Classification method	Dataset	Purpose	Limitations
[17]	TF-IDF	RBFN, PNN	Dataset of comments about two drugs in English	Two-class classification of comments (positive and negative)	Considering only the frequency of words in TF-IDF method, not considering neighbor words in the process of weighting.
[42]	Binary trees	Semi-supervised Recursive Auto encoder	Set of personal stories in English	Multi-class classification of English stories	Hard implementation
[43]	Time-Delay Neural Networks (TDNNs)	Recursive Neural Network	Dataset of comments about movie reviews in English	Two-class classification of comments (positive and negative)	Hard implementation
[15]	Word2vec, BOW, BOW+Word2-vec	Logistic Regression	Dataset of comments on various fields in English	Two-class classification of comments (positive and negative)	High reliance of logistic regression on a proper presentation of data (requires identifying all the important independent variables), vulnerable to overfitting.
[1]	Semantic dictionary + word2vec	SVM + Logistic Regression	Dataset of comments on films (ACLIMDB) in English	Two-class classification of comments (positive and negative)	Reduced accuracy and performance when the dimensions are very high (in text processing problems), re-training the SVM in case of any changes in cluster topology.
[10]	Dependency tree	LSTM+GRNN	Dataset of comments on various fields in English	Three-class classification of comments (positive, negative, neutral)	Relying on the Stanford dependency parser (which is not applicable for other languages).
[16]	Word2vec	Boosted CNN	Dataset of comments about films in English	Two-class classification of comments (positive and negative)	Sensitivity to noisy data and outliers of the boosted CNN.
[3]	Character-level n-gram model	NB, SVM and CNN	Costumers comments	Two-class	Not handling categorical variables

Table 1 (continued)

Reference	Feature selection method	Classification method	Dataset	Purpose	Limitations
			about laptops, books, and hotels	classification of comments (positive and negative)	and soft boundaries very well by the NB method.
[41]	Using a dictionary named Persianclues	LDASA	Costumers comments about cellphones and digital cameras in Persian	Two-class classification of comments (positive and negative)	In short texts, LDA models a document as mixture of topics, and then each word is drawn from one of its topic. You can imagine a black box contains tons of words generated from such a model. Now you have seen a short document with only a few of words. The observations is obviously too few to infer the parameters (the data sparsity problem).
[5]	MMI, MI, TFV, DF	Naïve Bayes	Dataset of costumers comments about various brands of cellphones in Persian	Two-class classification of comments (positive and negative)	Same disadvantage of NB in [3]
[8]	BOW	Lexicon -based	Dataset of online cellphone reviews written in Persian in two different website	Two-class classification of comments (positive and negative)	A positive or negative word may have opposite meanings in various fields. A sentence may contain several emotional words, but totally be neutral. On the other hand, a sentence may not contain any emotional words but overall contain a suggestion or sensation.
[2]	POS Tagger	Lexicon -based	Dataset of comments about one hotel in Persian	Three-class classification of comments (positive, negative, neutral)	Same as the disadvantages of the lexicon-based method in [8].
This work	Word2vec + CNN	Bi-LSTM	Digikala dataset, with 151229	Two and multi-class	Need of a large database for training

Table 1 (continued)

Reference	Feature selection method	Classification method	Dataset	Purpose	Limitations
			customer reviews about cellphones and digital cameras	classifica-tion of comments.	the BLSTM network.

model architectures including Continuous Bag-Of-Words (CBOW) model and continuous skip-gram model [33]. In Fig. 1, the structure of word2vec is shown. CBOW predicts target words from the text context. Context text is a sequence of words with a special window size:

$$Contex(w_j) = \{w_{j-c}, w_{j-c+1}, \ldots, w_{j-1}, w_{j+1}, w_{j+2}, \ldots, w_{j+c}\} \quad (1)$$

in which c denotes window size and w_j denotes target word. The window size is based on neighborhood distance. To predict the target word, CBOW should maximize $P(Context(w_j))$. To this end, the log-likelihood should be maximized:

$$l_{CBOW} = \frac{1}{k} \sum_{j=1}^{k} \sum_{-c \le i \le c, i \ne 0} log\, P(w_{j+i}) \quad (2)$$

where k denotes the number of training words that are a sequence such as w_1, w_2, …, w_k.

Skip-gram is similar to CBOW. The difference is that words in context text are predicted from one input word. The goal of skip-gram is to maximize $P(w_j)$. To this end, log-likelihood should be maximized:

$$l_{Skip\ gram} = \frac{1}{k} \sum_{j=1}^{k} \sum_{-c \le i \le c, i \ne 0} log\, P(w_j) \quad (3)$$

where $P(w_I)$ is computed using a softmax function as follows:

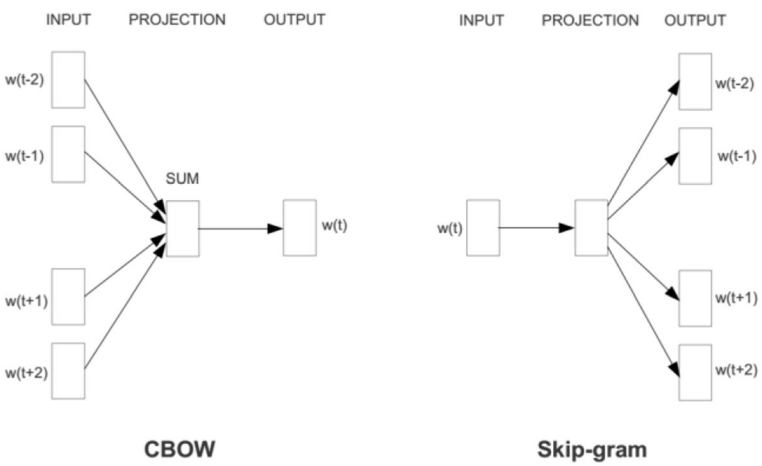

Fig. 1 Model of Word2vec [32]

$$P(w_I) = \frac{exp\left(V_{w_o}{}'^{T}.V_{w_I}\right)}{\sum_{w=1}^{W} exp\left(V_w{}'.V_{w_I}\right)} \tag{4}$$

such that V_w' and V_w denote the input and output vectors for the word w, and W denotes the number of words in the vocabulary.

3.2 Convolutional neural networks (CNN)

Convolutional neural networks are one of the most important deep learning algorithms. CNNs have many usages in the field of image processing and nowadays, they are used in the field of NLP as well. In addition, CNN mainly focuses on extracting the local features of the text; the fact that attention mechanisms are concerned with information in the context. Each CNN consists of three main layers [4, 49]:

1) Convolutional Layer: In this layer, several filters are used to convolve the input matrix. Each filter in the convolution layer maps the input matrix to another space and the output dimension of each filter depends on the dimension of the filter.
2) Pooling layer: This layer is usually placed after the convolutional layer. It is used to reduce the dimension of mapped features and reduce parameters. Functions such as average pooling and max pooling are used in this layer.
3) Fully connected layer: This layer is placed after the last pooling layer. It is similar to layers in the multilayer perceptron and the activation function in each layer is determined based on the network goal, such as classification or forecasting.

3.3 Long short term memory (LSTM)

LSTM is one kind of recurrent neural networks that was presented in 1997 by Hochreiter and Schmidhuber [19]. The simplest RNN cannot remember information for a long period of time, while LSTM is designed for long-term dependency that exists in certain types of data such as audio, video, and text. In LSTM, instead of a neuron with one activation function, the memory block is located in the hidden layer. Each memory block contains 3 gates: input gate, output gate and forget gate [18, 48]. A memory block is shown in Fig. 2. Using these gates, a memory block learns new information and forgets old information. Current block state is shown with C_t. The current block state is updated with old block state C_{t-1} and new information. The output of the hidden layer with block memory is calculated as follows:

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)\\
f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)\\
C_t^{\sim} &= tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)\\
C_t &= f_t{\cdot}C_{t-1} + i_t.\ C_t^{\sim}\\
h_t &= o_t.tanh(C_t)
\end{aligned} \tag{5}$$

where i denotes an input gate that adds new information to block memory, f denotes a forget gate that determines how much of the existing memory is forgotten, and o denotes an output gate that determines the amount of memory content. Ct and $C_t^{\sim}$ denote previous memory and

new memory, respectively, and together, they update the block state, and $\sigma(.)$ *and* $tanh\,(.)$ denote sigmoid and hyperbolic tangent functions, respectively. The operator . denotes element-wise multiplication.

4 The proposed method

In this section, the proposed method is described. It consists of several main modules and each module is described in the following subsections. Figure 3 shows a schematic flowchart of the proposed method.

4.1 Sentences preparation into the input matrix module

A text document may contain several sentences; each sentence contains numerous words, and each word contains several characters. For sentences to be understood by the computer for processing, it is necessary to have a preprocessing and mapping of the text to numerical values [20]. In sentences preparation into the input matrix module, the sentences are mapped to numerical values in two levels of the word and the character.

4.1.1 Word-level

Preparation of sentences into the input matrix module in word-level includes several steps. Each step is briefly explained.

- **Initial preprocessing:**

Preprocessing is an important process in the field of machine learning. In this paper, the "hazm" library in Python was used for preprocessing as a standard library in Persian language.

Initial preprocessing steps included:

– **Removal of punctuation marks:** The first step in preprocessing text documents is to remove punctuations. The punctuations removed included the following characters:

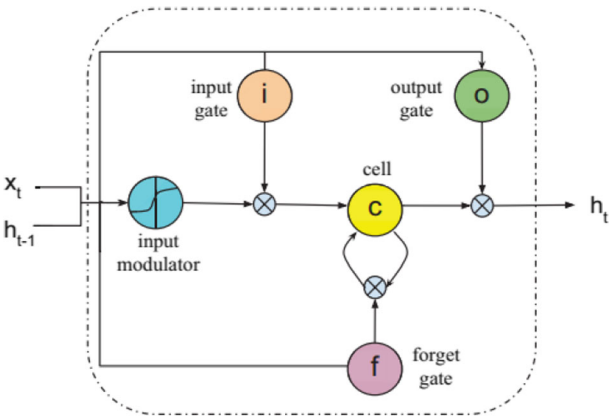

Fig. 2 Memory block

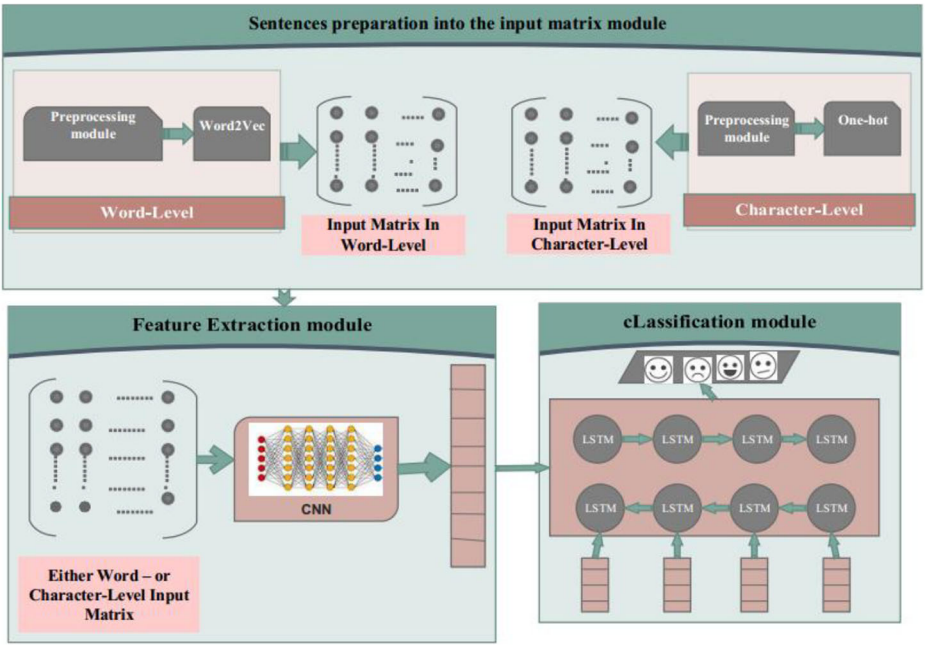

Fig. 3 The process of the proposed method

"!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~1234567890 ".

- **Normalization:** Some characters appear in Persian and Arabic languages in different forms. Also there are two types of distances in the Persian language. The half distance (zero-width non-joiner) is used for attaching prefixes and suffixes within a word and a complete distance is used to separate words. Table 2 shows examples of normalization in Persian language.
- **Stemming:** stemming is done to reduce the number of words (dimension reduction), computational complexity and running time. Stemming algorithms work by removing the suffix and prefix of the words.
- **Stop-words removal:** First, the list of stop words is collected and then, they are removed. These words usually have a high number of frequency but do not contain any specific meaning. By removing stop-words, we can concentrate on the words with more important meaning. Some examples include از (from), به (to), در (in), آن (that), تا (until), اکنون (now), این (this), ون (because), ….
- **Tokenization:** This step splits the documents into words/terms and converts each document to bag-of-words. This separation is done using unigrams, bigrams and trigrams, which were defined as sequences of one, two or three adjacent words from a list of tokens. We mostly used unigrams (similar to other studies), except some situations which need two-word expressions.

- **Secondary preprocessing:**

Informal texts in Persian face numerous challenges (Table 3). In every language, we can talk both formal and informal. Some users talk informally on the website and social networks. To solve one of the challenges, informal texts should become formal. To do this, there are no regular methods, but several methods are used to do this: stemming using different algorithms, using specific rules and using databases with pairs of informal and formal words. In this work, a combination of manual and automatic (using a database list) methods are used, in which a list of informal expressions and verbs are made by the authors with their corresponding formal equivalents.

- **Word Representation:**

In this step, first a large amount of text data for word2vec training is used, which was extracted from Wikipedia. Then words of each sentence are mapped to the d-dimensional vector using word2vec. For example, the sentence" اندازه و بزرگ است / The size of the phone is large" has 4 words in Persian. Hence, this sentence is mapped to a 4-dimensional vector. Then each sentence can be represented as a matrix. In this matrix, the number of rows is equal to the number of sentence words and the number of columns is equal to the vector's dimensions of each word (d).

4.1.2 Character-level

For preparation of sentences to the input matrix module in character-level, first all sentences are separated at the character level. Each character is represented by the one-hot vector. The one-hot vector length for each character is equal to the number of alphabets, and the vector consists of 0 s in all cells with the exception of a single 1 in a cell used uniquely to identify the character. The number of rows in this matrix is the number of sentence characters and the number of columns is the number of the alphabet of the language. This matrix is given as the input to the CNN.

4.2 Feature extraction module

The output of the previous step is given to the convolutional neural network. The convolutional neural network has several layers of convolution and pooling. In each layer of convolution, filters with different sizes are applied to the input. In the pooling layer, k-max pooling function is used.

The output vector of the second pooling layer is expressed as the vector of the input sentence. In the convolutional layer, filters are applied to the input matrix. The width of each filter should be the same as the dimension of the vectors (d). Each filter extracts a vector and the length of the extracted vectors depends on the length of the filters. In the pooling layer,

Table 2 Examples from Normalization

Before Normalization	After Normalization	English Translation	Details
تاب ها	تاب‌ها	Books	"ها" is a suffix
م بن	مب ن	I see	"م" is a prefix
با ي	با	Game	"ي "convert to " "
اع ا	اع ا	Premium	Removed " " from end of word

Table 3 Examples of informal texts

Informal	Formal	English Translation
.مـ هـ و خون ت	.مـ و د موبـا خان‌هـ ات	She says her mobile phone is home.
دو بن و عااااا هت	دو بن موبـا عا ات	Cellphone camera is excellent
.من مخا ب ا ن و بخ	.من مخواه بو ان و ابخ	I want to go to buy this phone.

using the k-max pooling function, all the extracted vectors in the convolution layer are mapped to a vector. This vector as input is given to the Bi-LSTM for classification.

4.3 Classification

In this step, the vectors that are prepared for each sentence by the previous module are given to the Bidirectional Long Short Term Memory (Bi-LSTM) network for sentiment classification. Bi-LSTM is similar to LSTM in terms of structure since it has memory blocks. Bi-LSTM has two parallel layers that extend in two directions: forward and backward. The outputs of two parallel layers are calculated as follows:

$$\begin{aligned} h_{f_t} &= H\left(W_{xh_f}X_t + W_{h_fh_f}h_{f_{t-1}} + b_{h_f}\right) \\ h_{b_t} &= H\left(W_{xh_b}X_t + W_{h_bh_b}h_{b_{t-1}} + b_{h_b}\right) \end{aligned} \tag{6}$$

where h_f and h_b denote the output vectors of forward and backward layers, respectively. The output of this network is $y_t = \left[h_{f_t}, h_{b_t}\right]$ that is the combination of h_{f_t} and h_{b_t}. To learn each token in a sequence, Bi-LSTM incorporates the previous and next contexts. In the last layer, the activation function is Softmax function. To predict the class label y = {1,…C} for sentence X, the Softmax function is used:

$$P_K = p(y = k \mid x, W) = \frac{e^{W_k^T X}}{\sum_{j=1}^{C} e^{W_j^T X}} \tag{7}$$

where P_K represents the probability of sentence X relative to class label k and W denotes weights of the layer related to the Softmax function.

In this work, the vector of each sentence is classified as a binary and a multi-class problem. Sentences in binary classification are divided into positive and negative classes. In multi-class classification, sentences are divided into five classes where each class is a numerical score. The higher (lower) scores represents more positivity (negativity).

5 Experimental results and comparison of the proposed method

In this section, we first present some of the simulation results of our proposed method and next intend to compare them with other methods in terms of efficiency and accuracy. To evaluate the proposed method, a dataset that contains a set of users comments on social sites is required. Users usually express their opinions in terms of numerical scores and some of the text comments. We used one of the most visited Persian sites, namely Digikala online store [13], launched in 2006 where customers express their opinions about the products. In this paper, to evaluate the proposed method the comments about cameras, cellphones and computer peripherals are used which are collected in the period between July 2016 and February 2017. A

customer comment is shown as an example in Table 4. These comments are labeled as binary (positive-negative) and multi-class (5 classes). Therefore, classification is done in two classes and in multi-class. The distribution of user comments in 5 classes is shown in Table 5.

As seen in Table 4, user comments have been rated from various aspects. However, these scores are not usable for our work since scores are in the document level. Therefore, to review comments at the sentence level, we used the labeled dataset (labels on sentences) which was done by Basiri and Kabiri in [6].

The performance of the classification algorithm is investigated by the following measures:

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

where:

- TP refers to the number of sentences correctly classified that belong to a special category C, for {1, …, C} classes.
- FN refers to the number of sentences incorrectly classified but belong to the category C, for {1, …, C} classes.
- FP refers to the number sentences incorrectly classified in category C, for {1, …, C} classes.

In addition, the F-measure is used which is a combination of the standard recall and precision measures, calculated as:

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision} \tag{10}$$

Table 4 A customer comment

Persian	English
به نظر من این گوشی با توجه به قیمتی که دارد خیلی گوشی خوبی است. دوربینش خیلی خوبه و از نظر ابعاد کاربرپسند است. شاید تنها عیب آن نداشتن زبان فارسی است.	I think regarding its price, it is a good cellphone. Its camera is very nice and user-friendly in terms of dimensions. Perhaps the only shortcoming is not supporting the Persian language.

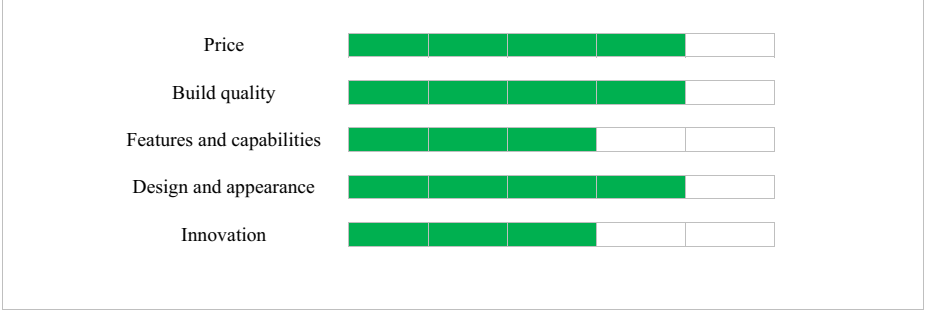

Table 5 Number of sentences in each class of Dataset

Class Name	Number of sentences	Sentiment level
Class 1	7521	Very bad
Class 2	10227	Bad
Class 3	102439	Moderate
Class 4	16040	Good
Class 5	15002	Excellent

In Table 6, feature extraction methods and deep learning algorithms for classification have been compared on the dataset with binary labels. The results show that the proposed method in word-level with 95% accuracy, 94% precision, 95% recall and 94% F-measure has the highest performance. Generally, according to the results, accuracy at both levels of the character and the word are close in this dataset and combining CNN and BLSTM outperforms using each algorithm individually.

To calculate evaluation measures for the dataset with multi-class labels, the confusion matrix should be computed. In Tables 7 and 8, the confusion matrices are calculated in two levels (word-level and character-level) for the dataset used in this paper.

The proposed method was run with different values for Bi-LSTM network parameters. The results in Tables 9 and 10 show that a higher number of cells in hidden layers usually leads to a better accuracy. The average accuracy obtained is 83% in the binary labeled dataset and 80% in the multi-labeled dataset. Furthermore, feature vector size in word2vec is an important parameter that affects classification accuracy in word-level. The value of this parameter should not be too low or too high. The results show the best dimension sizes are 250 and 200 for binary labeled and multi-labeled datasets, respectively.

Figures 4 and 5 show the effect of the number of filters on the convolutional layer in the convolutional neural network. The number of filters in the convolutional layer is actually the number of feature vectors that enter the Bi-LSTM network. Therefore, determining the number of convolutional layer filters is important. Results show that increasing the number of filters to a certain threshold increases accuracy, recall, precision and F-measure.

In addition to the parameters mentioned, other parameters should be set suitably. These parameters are set after initial experiments. Some of these parameters are as follows:

- Learning rate: 1E-3

Table 6 Comparison of proposed algorithm with previous algorithms

Feature extraction + deep learning method	Precision	Recall	F-measure	Accuracy
Word2vec + BLSTM	0.88	0.86	0.87	0.88
Word2vec + CNN	0.87	0.85	0.86	0.85
Word2vec + CNN + LSTM	0.92	0.94	0.93	0.93
Word2vec + CNN + BLSTM	0.94	0.95	0.94	0.95
Character-level + BLSTM	0.89	0.88	0.88	0.89
Character-level + CNN [3]	0.87	0.86	0.86	0.86
Character-level + CNN + LSTM	0.92	0.93	0.92	0.91
Character-level + CNN + BLSTM	0.95	0.94	0.94	0.94
CNN + Adaboost [16]	0.81	0.84	0.82	0.82

Table 7 Confusion matrix in word-level

	Class 1	Class 2	Class 3	Class 4	Class 5
Confusion matrix of word2vec + CNN					
Class 1	1069	111	103	110	111
Class 2	94	1641	98	112	100
Class 3	589	639	17968	640	651
Class 4	94	83	99	2830	102
Class 5	86	99	115	88	2612
Confusion matrix of word2vec + BLSTM					
Class 1	1106	103	116	86	93
Class 2	97	1658	94	104	92
Class 3	831	784	17228	802	842
Class 4	97	108	104	2801	98
Class 5	108	94	101	103	2594
Confusion matrix of word2vec + CNN + BLSTM					
Class 1	1343	38	37	42	44
Class 2	44	1844	50	53	54
Class 3	537	546	18275	571	558
Class 4	57	67	62	2972	50
Class 5	50	76	63	51	2760

- Batch size: Batch size defines number of samples that are going to be propagated through the network that in this work is set to 100.
- Number of iterations: 200

Table 8 Confusion matrix in character-level

	Class 1	Class 2	Class 3	Class 4	Class 5
Confusion matrix of Character-level + CNN					
Class 1	1090	95	98	118	103
Class 2	102	1636	96	113	98
Class 3	751	752	17486	776	722
Class 4	190	167	168	2494	189
Class 5	158	164	148	163	2367
Confusion matrix of Character-level + BLSTM					
Class 1	1121	95	95	85	108
Class 2	97	1654	101	106	87
Class 3	634	678	17848	646	681
Class 4	109	121	85	2801	92
Class 5	87	107	101	113	2592
Confusion matrix of Character-level + CNN + BLSTM					
Class 1	1341	53	33	38	39
Class 2	56	1809	62	59	59
Class 3	553	605	18116	616	597
Class 4	61	57	54	2977	59
Class 5	84	81	75	70	2690

Table 9 The effect of setting various parameters in binary labeled dataset

Feature vector size in word2vec	LSTM cell size	Training accuracy	Testing accuracy
50	5	0.88	0.74
50	10	0.90	0.77
50	15	0.91	0.78
100	5	0.90	0.79
100	10	0.93	0.83
100	15	0.92	0.84
100	20	0.93	0.83
150	5	0.95	0.84
150	10	0.94	0.83
150	15	0.95	0.86
200	20	0.97	0.90
200	20	0.96	0.93
250	20	0.99	0.95
250	15	0.95	0.88
250	20	0.93	0.88
300	20	0.89	0.87
350	10	0.88	0.78
350	15	0.89	0.79
350	20	0.85	0.76

6 Conclusion and future works

In this article, the design and implementation of a system for sentiment analysis of user opinions on Digikala's website was investigated. Previous works revealed that the traditional machine learning methods cannot understand semantic concepts precisely as a result of text integrity. The proposed method in this paper uses deep learning algorithms to enhance system learning ability. In the proposed method, preparation

Table 10 The effect of setting parameters in multi-labeled dataset

Feature vector size in word2vec	LSTM cell size	Train accuracy	Test accuracy
50	5	0.80	0.70
50	10	0.83	0.71
50	15	0.87	0.74
100	5	0.85	0.78
100	10	0.86	0.80
100	15	0.90	0.82
100	20	0.87	0.80
150	5	0.90	0.83
150	10	0.94	0.88
150	15	0.93	0.90
200	20	0.94	0.92
200	20	0.92	0.87
250	20	0.86	0.76
250	15	0.83	0.70
250	20	0.85	0.71
300	20	0.80	0.69

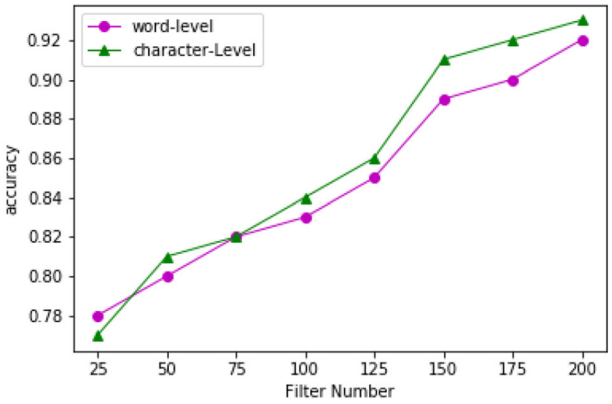

Fig. 4 Effect of filter number on Binary label dataset

of sentences into the input matrix is done separately on two levels: word-level and character-level. In word-level, words are mapped to the vector space by word2vec and then for each sentence, the generated vectors for each word create the matrix. Also, at the character-level, each character is represented by a numeric vector. Matrices of each sentence on both levels are given separately to the convolutional neural network as the input. The convolution neural network output is a numeric vector for each sentence in the vector space. Finally, the generated vectors for each sentence are classified using the Bi-LSTM network. Results of simulations on Digikala's users comments are compared with two other studies and five combinations of feature extraction and deep neural networks. The accomplished results show that the proposed method has a high accuracy for binary classification and multi-label classification and outperforms other methods. The accuracy reached by the proposed algorithm is 95% in the word-level and 94% in the character-level for two classes. On the other hand, the proposed method has the limitation of requiring a large dataset for training the BLSTM and high complexity in the character level.

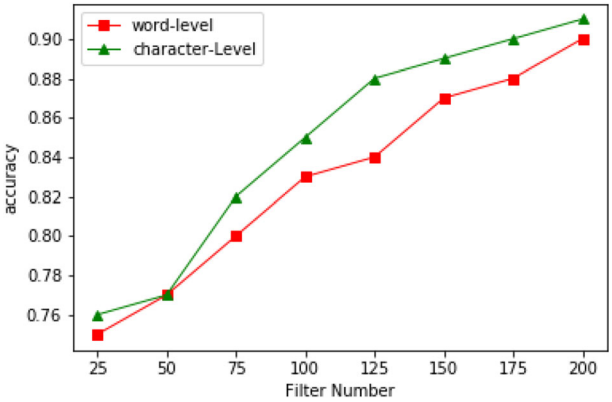

Fig. 5 Effect of filter number on Multi-label dataset

With regard to future studies, we suggest to implement deep learning algorithms on big data frameworks such as: Hadoop, Spark, etc., since it can simultaneously benefit from deep learning algorithms and big data frameworks and provide good performance in data processing.

Acknowledgements The authors would like to thank Shahid Chamran University of Ahvaz High PerformanceComputing Center (SCU-HPCC) for providing computing resources for this project.

References

1. Alshari EM, Azman A, Doraisamy S, Mustapha N, Alkeshr M (2017) Improvement of sentiment analysis based on clustering of Word2vec features. In: 28th international workshop on database and expert systems applications (DEXA)
2. Amiri F, Scerri S, Khodashahi M, Iais F, Augustin S (2015) Lexicon-based sentiment analysis for Persian text, pp 9–16
3. An Y, Tang X, Xie B (2017) Sentiment analysis for short Chinese text based on character-level methods. In: 9th international conference on knowledge and smart technology (KST)
4. Anwar S, Hwang K, sung W (2015) Fixed point optimization of deep convolution neural networks of object recognition. ICASSP, pp 1131–1135
5. Bagheri B, Saraee M, de Jong F (2013) Sentiment classification in Persian: introducing a mutual information-based method for feature selection. In: 2013 21st Iranian conference on electrical engineering (ICEE), pp 1–6
6. Basiri ME, Kabiri A (2017) Sentence-level sentiment analysis in Persian. 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), Shahrekord, Iran, pp 84–89
7. Basiri M, Nilchi A, Ghassem-Aghaee N (2014) A framework for sentiment analysis in Persian. Open Transactions on Information Processing 1:1–14
8. Basiri M, Naghsh-Nilchi A, Ghassem-Aghaee N (2014) A framework for sentiment analysis in persian. Open Trans Inf Process 1(3):1–14
9. Boudad N, Faizi R, Oulad Haj Thami R, Chiheb R (2017) Sentiment analysis in Arabic: a review of the literature. Ain Shams Engineering Journal 9:2479–2490
10. Chau N, Phan V, Nguyen M (2016) Deep learning and sub-tree mining for document level sentiment classification. In: Eighth International Conference on Knowledge and Systems Engineering (KSE)
11. De Mulder W, Bethard S, Moens M (2015) A survey on the application of recurrent neural networks to statistical language modeling. Comput Speech Lang 30(1):61–98
12. Dhomne A, Sa P (2018) Face verification using deep learning. JIMS8I International Journal of Information Communication and Computing Technology 6(1):332
13. Digikala (2017) [Online]. Available: http://www.digikala.com. Accessed: 15 Feb 2017
14. Elman J (1990) Finding structure in time. Cogn Sci 14(2):179–211
15. Enríquez F, Troyano JA, López-Solaz T (2016) An approach to the use of word embeddings in an opinion classification task. Expert Syst Appl 66:1–6
16. Gao Y, Rong W, Shen Y, Xiong Z (2016) Convolutional neural network based sentiment analysis using Adaboost combination. In: International Joint Conference on Neural Networks (IJCNN)
17. Gopalakrishnan V, Ramaswamy C (2017) Patient opinion mining to analyze drugs satisfaction using supervised learning. Journal of Applied Research and Technology 15(4):311–319
18. Graves A (2013) Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850
19. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
20. Krouska A, Troussas C, Virvou M (2016) The effect of preprocessing techniques on twitter sentiment analysis. In: 7th International Conference on Information, Intelligence, Systems & Applications (IISA)
21. Lang K, Waibel A, Hinton G (1990) A time-delay neural network architecture for isolated word recognition. Neural Netw 3(1):23–43
22. Lau R, Xia Y, Ye Y (2014) A probabilistic generative model for mining cybercriminal networks from online social media. IEEE Comput Intell Mag 9(1):31–43
23. Le QV, Mikolov, T (2014) Distributed representations of sentences and documents, computation and language
24. Lei X, Qian X, Zhao G (2016) Rating prediction based on social sentiment from textual reviews. IEEE Transactions on Multimedia 18(9):1910–1921

25. Li X, Xie H, Chen L, Wang J, Deng X (2014) News impact on stock price return via sentiment analysis. Knowl-Based Syst 69:14–23
26. Li H, Peng Q, Guan X (2016) Sentence level opinion mining of hotel comment. IEEE International Conference on Information and Automation (ICIA)
27. Li Q, Jin Z, Wang C, Zeng D (2016) Mining opinion summarizations using convolutional neural networks in Chinese microblogging systems. Knowl-Based Syst 107:289–300
28. Li X, Peng L, Yao X, Cui S, Hu Y, You C, Chi T (2017) Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation. Environ Pollut 231:997–1004
29. Liu B (2012) Sentiment analysis and opinion mining. Morgan & Claypool, San Rafael
30. Liu Z, Chen Y, Tang B, Wang X, Chen Q, Li H, Wang J, Deng Q, Zhu S (2015) Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. J Biomed Inform 58:S47–S52
31. McLeod P, Shallice T, Plaut D (2000) Attractor dynamics in word recognition: converging evidence from errors by normal subjects, dyslexic patients and a connectionist model. Cognition 74(1):91–114
32. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
33. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space, computation and language
34. Pang L, Zhu S, Ngo C (2015) Deep multimodal learning for affective analysis and retrieval. IEEE Transactions on Multimedia 17(11):2008–2020
35. Razzaghnoori M, Sajedi H, Jazani I (2018) Question classification in Persian using word vectors and frequencies. Cogn Syst Res 47:16–27
36. Rill S, Reinel D, Scheidt J, Zicari R (2014) PoliTwi: early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis. Knowl-Based Syst 69:24–33
37. Sainath T, Weiss R, Wilson K, Li B, Narayanan A, Variani E, Bacchiani M, Shafran I, Senior A, Chin K, Misra A, Kim C (2017) Multichannel signal processing with deep neural networks for automatic speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing 25(5):965–979
38. Sangeetha T, Balaganesh N, Muneeswaran K (2017) Aspects based opinion mining from online reviews for product recommendation. In: International conference on computational intelligence in data science(ICCIDS)
39. Saon G, Picheny M (2017) Recent advances in conversational speech recognition using convolutional and recurrent neural networks. IBM J Res Dev 61(4/5):1:1–1:10
40. Schouten K, Frasincar F (2016) Survey on aspect-level sentiment analysis. IEEE Trans Knowl Data Eng 28(3):813–830
41. Shams M, Shakery A, Faili H (2012) A non-parametric LDAbased induction method for sentiment analysis. In: The 16th CSI international symposium on artificial intelligence and signal processing (AISP 2012), pp 216–221
42. Socher R, Pennington J, Huang E, Ng A, Manning C (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, pp 151–161
43. Socher R, Huval B, Manning CD, Ng AY (2012) Semantic compositionality through recursive matrix-vector spaces. In: Proceeding EMNLP-CoNLL '12 proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp 1201–1121
44. Sun Y, Wang X, Tang X (2016) Hybrid deep learning for face verification. IEEE Trans Pattern Anal Mach Intell 38(10):1997–2009
45. Sun S, Luo C, Chen J (2017) A review of natural language processing techniques for opinion mining systems. Information Fusion 36:10–25
46. Wandabwa H, Asif Naeem M, Mirza F (2017) Document level semantic comprehension of noisy text streams via convolutional neural networks. In: IEEE 21st international conference on computer supported cooperative work in design (CSCWD)
47. Xu X, Cheng X, Tan S, Liu Y, Shen H (2013) Aspect-level opinion mining of online customer reviews. China Communications 10(3):25–41
48. Zaremba W, Sutskever I (2014) Learning to execute. arXiv preprint arXiv:1410.4615
49. Zhang Y, Wallace B (2015) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, computation and language.
50. Zhao R, Mao K (2017) Topic-aware deep compositional models for sentence classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing 25(2):248–260

Shima Zobeidi received her M.Sc. and B.Sc. degrees in Computer Engineering, major in Artificial Intelligence from Shahid Chamran University of Ahvaz (SCU), Ahvaz, Iran in 2018 and 2014, respectively. Her research interests include artificial intelligence, deep learning algorithms and libraries, distributed systems, fuzzy logic and evolutionary methods in networks.

Marjan Naderan received her B.Sc. degree in Computer Engineering in 2004 and the M.Sc. degree in Information Technology in 2006 both from Sharif University of Technology (SUT), Tehran, Iran. She received the Ph.D. degree in Computer Engineering, major in computer networks in Feb. 2012, from Amirkabir University of Technology (AUT), Tehran, Iran. Dr. Naderan joined the Computer Engineering department of Shahid Chamran University (SCU) in Ahvaz, Iran in Sep. 2012. She was the head of the Computer Engineering department from 2013 to 2015. She is currently the director of the HPC Center in Shahid Chamran University of Ahvaz (SCU-HPCC). Her research interests include computer networks, wireless and mobile networks, IoT and cloud computing, social networks, object tracking, network optimization, simulation of network protocols and bio-inspired and intelligent methods in networks.

Seyyed Enayatallah Alavi received his B.Sc. degres in Computer Engineering, major in Hardware, from Isfahan University of Technology, Isfahan, Iran, in 1992 and his M.Sc. degree in Computer Engineering major in Artificial Intelligence and Robotics from Shiraz University, Shiraz, Iran, in 1995. He also received his Ph.D. degree in Computer Engineering from National Belarusian University of Technology, Minsk, Belarus, in 2012. Since 2012, Dr. Alavi joined the Computer Engineering department in Shahid Chamran University of Ahvaz, Ahvaz, Iran, and he was the head of that department from 2012 till 2013. His research interests include Fuzzy logic, Neural networks, Bio-inspired algorithms and Image Processing.