IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 12, 2020, accepted April 26, 2020, date of publication May 4, 2020, date of current version May 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992063

Multi-View Deep Network: A Deep Model Based on Learning Features From Heterogeneous Neural Networks for Sentiment Analysis

HOSSEIN SADR[1], MIR MOHSEN PEDRAM[2], AND MOHAMMAD TESHNEHLAB[3]
[1]Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran
[2]Department of Electrical and Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran
[3]Industrial Control Center of Excellence, Faculty of Electrical and Computer Engineering, K. N. Toosi University, Tehran, Iran

Corresponding author: Mir Mohsen Pedram (pedram@khu.ac.ir)

ABSTRACT By the development of social media, sentiment analysis has changed to one of the most remarkable research topics in the field of natural language processing which tries to dig information from textual data containing users' opinions or attitudes toward a particular topic. In this regard, deep neural networks have emerged as promising techniques that have been extensively used for this aim in recent years and obtained significant results. Considering the fact that deep neural networks can automatically extract features from data, it can be claimed that intermediate representations extracted from these networks can be also used as appropriate features. While different deep neural networks are able to extract various types of features due to their distinct structures, we decided to combine features extracted from heterogeneous neural networks using multi-view classifiers to enhance the overall performance of document-level sentiment analysis by considering the correlation between them. The proposed multi-view deep network makes use of intermediate features extracted from convolutional and recursive neural networks to perform classification. Based on the results of the experiments, the proposed multi-view deep network not only outperforms single-view deep neural networks but also has superior efficiency and generalization performance.

INDEX TERMS Deep learning, multi-view learning, convolutional neural network, recursive neural network, sentiment analysis.

I. INTRODUCTION

By the rapid development of the World Wide Web, especially social media, a large amount of textual data containing people's opinions and feelings is generated. While this textual data is precious, useful and can be employed by companies, government, and other people for making decisions, there is a need to develop an intelligent system that can automatically extract valuable information from them and classify them based on their polarities. This issue is investigated in one of the fields of natural language processing known as sentiment analysis [1].

Although numerous studies have been carried out to propose accurate methods for the task of sentiment analysis in the last decades, the emergence of deep learning, as a subset of machine learning, has made a dramatic improvement in this field in recent years [2]. Deep learning enables computational methods that are made of numerous processing layers to learn various representations of data considering the different levels of abstraction. In fact, deep learning methods are able to utilize multiple processing layers to generate various valuable features from data without human intervention [3], [4]. Accordingly, these methods have made a dramatic improvement and impressive advancement in different fields, such as computer vision [5], speech recognition [6], and natural language processing [7]. Following this trend, various deep learning methods like convolutional neural network, recurrent neural network, and recursive neural network have been presented since the last previous decades and recent studies are now increasingly focusing on their new use and improvement [8], [9].

Although deep learning methods attracted a lot of attention and made considerable advancement, each of them has its potential and drawback and is able to extract particular types

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi.

of features from data. Having this ability in our mind, it can be indicated that intermediate representations that are extracted from deep neural networks can be suitable to be leveraged as features [10]. Otherwise, while various deep learning methods have different structures, they are able to generate different kinds of intermediate representations.

On the other hand, multi-view representation learning has recently attracted a lot of researchers and now is considered as one of the most promising directions in the field of machine learning and data mining [11]–[13]. Multi-view learning tries to learn the features of multi-view data with the aim of developing prediction models. In other words, multi-view learning employs heterogeneous properties of input data and compared to single-view learning is able to learn features on each view and train them jointly to enhance efficiency [14]. In this regard, we decided to investigate the combination of deep features extracted from heterogeneous deep neural networks to identify if they are complementary or they can be combined with a multi-view classifier to enhance the performance of textual sentiment analysis.

Briefly, in this paper, a sentiment analysis framework is proposed that uses heterogeneous deep neural networks to extract features of input documents and classify them using a multi-view classifier. The proposed multi-view deep network is applied to single-view data and tries to make use of their potentials. In this regard, firstly, abstract representations of words are learned from a large amount of data. Then, convolutional and recursive neural networks are simultaneously used to generate various representations of input text that are considered as intermediate features. The sets of features that are extracted by each deep neural network are considered as distinct views. Finally, a multi-view classifier processes each set that is regarded as a separate view and trains them jointly to determine the sentiment polarity of each document.

In general, the contribution of this paper is as follows: 1) Although multi-view learning has been extensively used in the field of image processing and obtained considerable results, it is still in its early steps of development in the field of natural language processing and only limited studies have been conducted focusing on document classification and to the best of our knowledge it is the first study that tries to employ multi-view learning for document-level sentiment analysis, 2) Multi-view methods are generally applied on more than one source of data and try to find agreement among them, while the proposed multi-view deep network is trained on one source and two different deep neural networks are used to extract to distinct views that are fed to multi-view classifier in parallel, 3) Employing multi-view classifier on the top of views extracted from deep neural networks provides this opportunity to make use of the potentials of convolutional and recursive neural networks at the same time with the aim of improving the overall classification accuracy. Finally, based on the results of the experiments, the proposed multi-view deep network obtained superior results compared to both traditional and combinational deep learning methods.

The remainder of this paper is classified as follows: Multi-view learning and related studies are introduced in section 2. The procedure of choosing intermediate representations form deep neural networks is explained in section 3. The details of the proposed multi-view deep network containing classical deep neural networks are described in section 4. Experimental details and obtained results are extensively indicated in section 5. Section 6 contains the conclusion and suggestions for future research in this filed.

II. RELATED WORK

Considering the fact that the proposed model of this paper tries to make use of both multi-view learning and deep neural networks, the literature review is also divided into two sections. The first section covers multi-view learning and remarkable studies that have been conducted in this filed. Deep neural networks and their applications in the field of sentiment analysis from the perspective of multi-view learning are reviewed in the second section.

A. MULTI-VIEW LEARNING

The goal of multi-view learning is to handle the challenge of learning features of multi-view data in order to extract appropriate information as well as providing a remarkable prediction model. In other words, developing entities in real-world using only one measuring technique is very challenging and may lead to the ignorance of some important aspects of them. However, data generated using multiple measuring techniques can help depict all sides of them [12], [14]. Considering the fact that multi-view data are extensively available in numerous real-world applications, multi-view learning, which is a new direction in machine learning that tries to improve generalization efficiency by learning multiple views, has attracted a lot of attention in recent years and has been extensively employed in various fields, such as cross-view classification [15], information retrieval [16], image classification [17], human pose estimation [18], and so on.

In general, most of multi-view learning methods use single-view algorithms for learning and try to find low dimensional common subspace to exhibit data. In this regard, multi-view learning methods are generally categorized into three major styles considering various applications of multi-view data [19]: 1) Co-training style algorithms, 2) Co-regularization style algorithms, and 3) Margin consistency style algorithms.

Co-training technique is one of the basic multi-view techniques which is based on single-view machine learning methods and is typically employed for semi-supervised learning. Co-training techniques are trained on two individual views while confident labels are used for unlabeled data and along with each iteration, unlabeled data of each view are labeled considering classifier prediction of the other view [19]. For instance, Co-training [20], Co-EM [21], and Co-clustering [22] are representative algorithms of this style.

Co-regularization techniques use regularization terms in the objective function to enhance the agreement between

classifiers of two views [14]. In fact, they try to clarify data if two distinct views are consistent or not. CCA [23], SVM-2K [24], MULDA [25], and MvDA [26] belong to this style of techniques.

Margin consistency algorithms are particularly introduced as the subset of Maximum Entropy Discrimination (MED) while margin is regarded as that in MED classifier. They are able to take advantage of the consistency classification results of two views. Algorithms in this category are based on this hypothesis that margin from two distinct views is consistent and therefore it can be stated that different views share identical classification confidence [14]. MVMED [27], SMVMED [28], and MED-2C [29] are algorithms that are classified in this style.

B. DEEP NEURAL NETWORK

While the focus of this paper is on using deep neural networks to extract features of distinct views, the application of deep neural networks in the field of sentiment analysis from the perspective of multi-view learning is explored in the following.

There is no doubt that in recent years deep learning has made a revolution in researches in the field of natural language processing, particularly sentiment analysis. Convolutional Neural Network (CNN) is one of the most studied deep learning methods in the field of sentiment analysis. Kim [30] conducted a series of experiments based on one layer convolutional neural network for this aim. Zhang *et al.* [31] presented a character-level convolutional neural network for text classification that showed significant enhancement in classification accuracy. Moreover, Kalchbrenner *et al.* [32] proposed a dynamic convolutional neural network that utilized dynamic k-max pooling.

Recurrent Neural Network (RNN) is another deep learning method that considers sequential data. In this regard, Tai *et al.* [33] employed Long Short Term Memory (LSTM) network integrated with some complex units for sentiment analysis. Kuta *et al.* [34] proposed tree structure gated recurrent neural network which was inspired by tree structure LSTM and adaptation of Gated Recurrent Unit (GRU) to recursive model. Besides these networks, a semi-supervised model, known as the Recursive Neural Network (ReRNN), has been also employed for the task of sentiment analysis which uses continuous word vectors as input and hierarchical structure. In this regard, Socher *et al.* [35] introduced a model, known as MV-RNN, that employed both matrix and vector with the aim of representing words and phrases in the tree structure.

From the perspective of multi-view learning, autoencoder is an unsupervised deep neural network that has been extensively employed. Ngiam *et al.* [36] proposed a bimodal deep autoencoder for extracting features. They used consecutive final hidden coding of video and audio as the inputs and these inputs were then mapped to share representation. Multi-view convolutional neural networks have also achieved considerable attention in speech recognition, machine vision, and natural language processing. Following a similar line of research, Su *et al.* [37] proposed a framework based on the multi-view convolutional neural network to unify and compact features that were extracted from two various views. In the following, Feichtenhofer *et al.* [38] also explored different methods to fuse representations obtained from convolutional neural networks temporally and spatially to extract informative features for recognizing human actions in the video. Recurrent neural networks have been also used for this aim. Cho *et al.* [39] employed recurrent neural network to present RNN encoder-decoder model using multi-view learning. Moreover, Sutskever *et al.* [40] used the LSTM network to propose a multi-view sequence to sequence learning. Multi-view recurrent neural networks have been also widely applied to various applications like information retrieval [41], video and image captioning [42], [43], visual question answering [44], and so on.

Otherwise, multi-view learning has been also used in the field of sentiment analysis and opinion mining. In this respect, multi-view learning was generally applied to social media consisting of pictures, videos, and text while each of them is regarded as an individual view. From this point of view, Tang *et al.* [45] presented a multi-view model that was based on the relation among views to choose the most related features. Niu *et al.* [46] also reported a comprehensive introduction to the models on this subject. Wan [47] also employed a combination of co-training and machine learning techniques to take advantage of unlabeled data in another language. Huang *et al.* [48] also used margin consistency and co-regularization techniques combined with deep neural networks for opinion mining of text.

Although deep neural networks and multi-view learning have been extensively utilized in recent years, it can be claimed that the motivation and the used methodology of this paper that tries to combine features extracted from the heterogeneous deep neural networks are entirely different from other state-of-the-art. To the best of our knowledge, it is the first time that the combination of features extracted from convolutional and recursive neural networks are fed to a multi-view classifier while the effect using various multi-view learning techniques in the family of co-regularization and margin consistency algorithms is also explored.

III. EXTRACTING FEATURES USING DEEP NEURAL NETWORKS

Extracting appropriate features is considered as a significant step in various applications of natural language processing and extensive studies have been carried out on designing robust features. Following a similar line of research, much attention has been given to feature engineering. Compared to traditional neural networks, deep neural networks do not require predefined features and instead, they can learn specific features during the training process. This process has made them a good option to be used as a feature extractor. To this end, a comprehensive introduction of two typical deep

neural networks and the motivation behind using them as feature extractors are provided in the following sections.

A. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network is known as one of the most famous types of deep neural networks which is designed to provide a suitable representation of the input. In fact, the strength of deep learning compared to machine learning refers to its ability to extract features without requiring human intervention and convolutional neural network, according to its structure, can be considered as a good option for this aim. Convolutional neural network generally contains convolutional layers, pooling layers, and several fully connected layers on top. *Convolutional layer* is started with the word matrix as an input where each row demonstrates the representation of each token. By repeatedly applying various convolutional filters on the input matrix, various feature maps indicating valuable patterns of input data are produced. Due to the diversity of feature maps achieved from various filter sizes, a *pooling layer*, typically max pooling, is required to induce fixed size vectors by finding the maximum from the local features of previous outputs. This manipulation aims to capture the significant features and reduce dimensionality. The output is then fed to a *fully connected neural network* on top that employs generated features to complete a classification task [49].

- **CNN Features:** It is worth mentioning that convolutional and pooling layers lead to the production of intermediate representations of input data, known as low-level local features, which are essential for consecutive computations. Therefore, despite machine learning methods that employ handcraft or raw features that are usually meaningless and ineffective, deep neural networks are able to learn representation automatically which is also notable in convolutional neural network while its structure is particularly designed for hierarchical feature extraction. Overall, considering the fact the fully connected layer on top of the convolutional neural network works like a traditional feed-forward neural network, it can be stated that intermediate variables between convolutional and pooling layers can be employed as automatically extracted features that are efficient and purposeful.

B. RECURSIVE NEURAL NETWORK

Recursive neural network is another type of deep neural networks that takes sequential data as an input and compute compositional vector representation of various length phrases which can be employed as features required for classification. In other words, n-grams are given as input to the recursive neural network, and based on the compositional model they are parsed into a binary tree where each leaf node corresponds to the vector representation of a word. In the following, by employing various kinds of compositionality functions, parent vectors are computed in a bottom-up fashion and then fed to the classifier as features. In other words, a recursive neural network consists of an architecture in which by having a positional directed cyclic graph, nodes are visited in topological order and transformations are applied recursively to produce more representations from the previously computed representation of children [50].

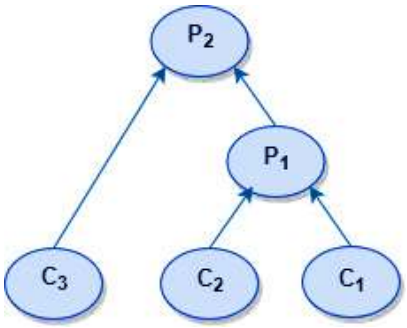

FIGURE 1. Structure of a typical recursive neural network.

A typical structure of a recursive neural network is illustrated in Figure 1. Considering that C_1 and C_2 are word representations of input words, P_1 is the parent vector having two children which is computed using $f(\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} + b)$ where f is the nonlinearity function. By applying this function recursively, vectors for multiword sequences can be obtained [50].

- **ReRNN features:** As it is clear, the information in recursive neural network travels from one node to another in a parse tree where information of parent vector is achieved through implicit interaction of children. Therefore, it can be assumed that the information passed between nodes carries important features of input sequences that are not only different from the convolutional neural network's extracted features but also contain valuable information that can be used for classification.

IV. PROPOSED MULTI-VIEW DEEP MODEL

The proposed multi-view deep network contains four separate components. Firstly, input data are processed to provide vector representations of words that are used in steps afterward. Next, convolutional and recursive neural networks are trained in parallel and act like feature extractor. In the following, intermediate representations extracted from both deep neural networks are passed to a multi-view classifier as two separate feature sets to perform classification. The overall flowchart of the proposed model is depicted in Figure 2.

A. WORD EMBEDDING

While documents are written in natural language, presenting them in the form to be understandable for deep neural networks is very challenging. Although, traditional one-hot vectors have been extensively employed in this field, using word vectors instead of them has efficiently improved the performance of many natural language processing tasks [9]. In this regard, *Word2vec* which is a shallow two layers neural

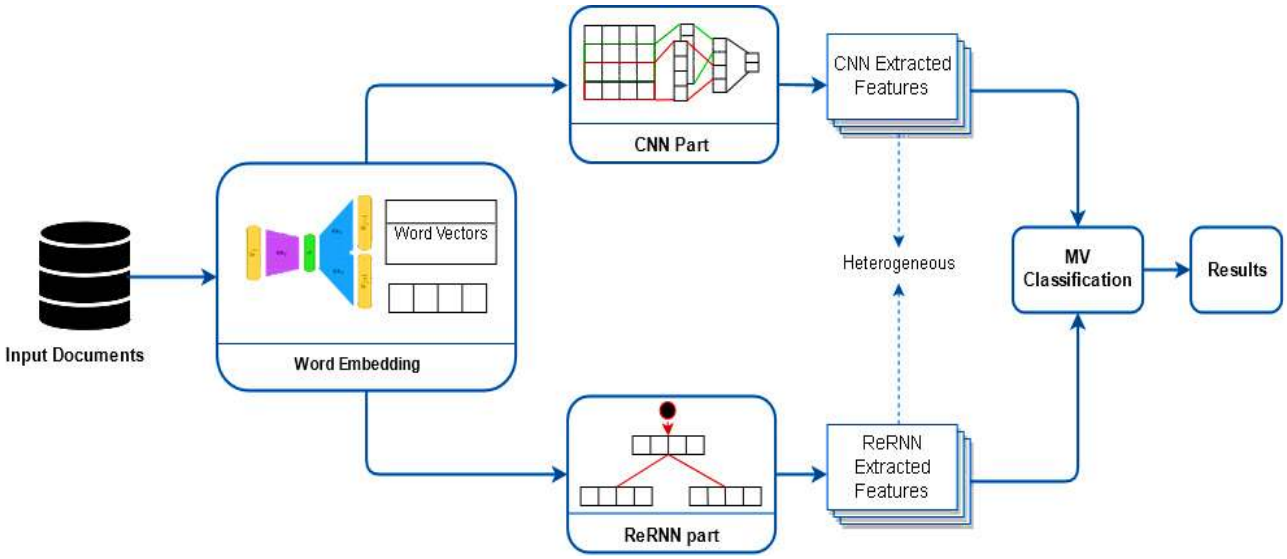

FIGURE 2. The overall flowchart of the proposed model containing four components of word embedding, CNN part, ReRNN part and multi-view (MV) classification.

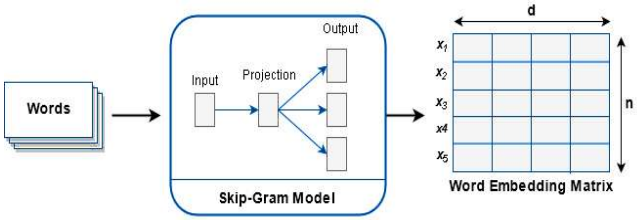

FIGURE 3. The flowchart of word embedding part.

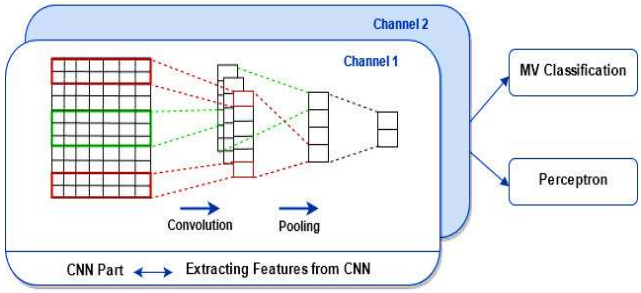

FIGURE 4. The flowchart of convolutional neural network part.

network is employed as a primary part of the proposed model to convert words into D-dimensional word vectors. *Word2Vec* is available in two different versions, a model based on Continuous Bag of Words (CBOW) and a model based on Skip-Gram which is used in this paper [51]. Generally, Skip-Gram learns the vector representation of a word by considering its context. The flowchart of the word embedding section using Skip-gram is presented in Figure 3.

As the Skip-gram training process is finished, results are saved as a look-up table where each word has its corresponding vector. Considering that n is the number of input words and d is the length of word embedding, each word is therefore encoded by a row vector in embedding matrix $A \in \mathcal{R}^{n\times d}$ which is considered as a single-view data for a document. This single-view data is then passed into convolutional and recursive neural networks respectively.

B. EXTRACTING FEATURES FROM CNN

To produce new features from the input text, the convolutional operation must be applied on sentence matrix $\in \mathcal{R}^{n\times d}$. To generate a variety of features, several channels with various window sizes are considered in our model. In each channel, data are divided into different pieces and are then processed using convolutional operation, activation function and pooling layer. Each channel also contains an exclusive window size known as a filter. While the sequential structure of a sentence has a prominent impact in determining its meaning, each filter width is supposed to be equal to the dimensionality of word vector (d) and filter weight (h) can be varied. Therefore, by considering that each filter is described by matrix $w \in \mathcal{R}^{h\times d}$, the output sequence ($O_i = woA[i : i + h - 1]$) is obtained by repeatedly applying filter (w) on sentence matrix (A). Where $i = 1, \ldots, s-h+1$ and o is the dot product between the inputs matrix and convolution filter. By adding a bias term ($b \in \mathcal{R}$) and *Relu* activation function, new features are generated. In the following, by applying the pooling layer, max pooling for instance, the most important features indicating the pattern of input documents are obtained [52]. Generally, the results of various channels are concatenated and then passed to the perceptron. The overall structure is presented in Figure 4.

The idea behind our model is that instead of passing the output features to perceptron, the extracted intermediate variables can be considered as an individual view. The reason is twofold: Firstly, while the main part of convolutional neural network structure is finished, it can be stated that the results in this stage contain its most characteristic operation. Secondly,

calculation related to perceptron is linear while it is one of the most basic tasks in machine learning. Therefore, it can be concluded that while perceptron is able to perform this task efficiently, extracted intermediate variables are then appropriate features for classification. Overall, extracted intermediate variables are considered as the first exclusive view which is going to be trained using the supervised multi-view classifiers in the following steps.

C. EXTRACTING FEATURES FROM ReRNN

The recursive neural network that is used in our model is basically the one in [53] which tries to fit a more complex compositionally function. This model adds new features to the general recursive neural network through a nested neural layer. In other words, to compute parent vector, firstly, a new feature is computed using the nested neural layer and by employing this new feature along with children's vectors, the parent's vector representation is obtained and each node employs Softmax classifier to predict its label. Considering that V_{c1} and V_{c2} are children's vectors and V_m is the new feature that is computed using $V_m = f(w^1 \begin{bmatrix} V_{c1} \\ V_{c2} \end{bmatrix} = b)$. Therefore, V_p will be the parent vector that is computed using $V_p = f(w^2 \begin{bmatrix} V_{c1} \\ V_{c2} \\ V_m \end{bmatrix} = b)$. The overall structure is presented in Figure 5.

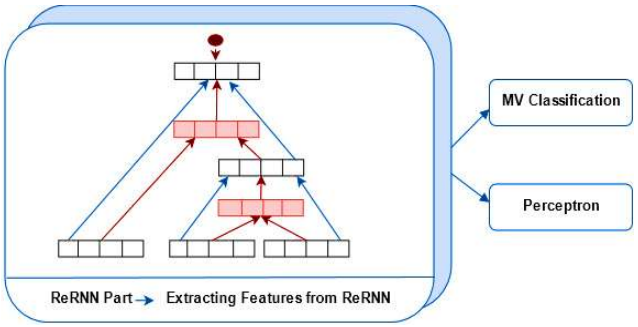

FIGURE 5. The flowchart of recursive neural network part.

Following the similar idea mentioned in the previous section, it can be stated that intermediate features extracted in this stage contain its most characteristic operations. Consequently, extracted intermediate variables can be also considered as the second exclusive view which is going to be trained using the supervised multi-view classifiers in the following steps.

D. MULTI-VIEW CLASSIFICATION

Intermediate features extracted from previous layers are considered as two separate views. Therefore, by having two various views of documents and their sentiment labels, the multi-view classifier can be trained to fit them. In other words, by applying multi-view learning, heterogeneous properties of the dataset can be utilized and by learning a function on each view and training them jointly, the overall performance can be enhanced. In this regard, among supervised multi-view classifiers, KCCA [23] and SVM-2k [24] in the group of co-regularization style algorithm and MVMED [27] and SMVMED [28] in the group of margin consisting style algorithm are chosen and explained in details in the following.

1) KCCA ALGORITHM

Canonical Correlation Analysis (CCA) is one of the representatives of co-regularization style algorithms that tries to correlate linear relationships between two various feature sets [23]. CCA aims to find a linear transformation for each feature set and then maximize the correlation between these transformed feature sets. In the following, the covariance of each transformed feature set is regularized to have enough small value. Considering that there are two feature sets demonstrating each view, CCA seeks to compute two projection directions w_A and w_B corresponding to the first and second views respectively and maximize the linear correlation coefficient as follows (Eq.1):

$$\frac{cov(w_A^T X, w_B^T Y)}{\sqrt{Var(w_A^T X)Var(w_B^T Y)}} = \frac{w_A^T C_{AB} w_B}{\sqrt{(w_A^T C_{AA} w_A)(w_B^T C_{BB} w_B)}} \quad (1)$$

where C_{AB}, C_{AA} and C_{BB} are the covariance matrices which are calculated as $C_{AB} = \frac{1}{n} XY^T$, $C_{AA} = \frac{1}{n} XX^T$ and $C_{BB} = \frac{1}{n} YY^T$. While w_A and w_B are scale-independent parameters, the following optimization (Eq.2) can be obtained.

$$\begin{aligned} &max_{w_A, w_B} = w_A^T C_{AB} w_B \\ &\text{Such that } w_A^T C_{AA} w_A = 1, \quad w_B^T C_{BB} w_B = 1 \end{aligned} \quad (2)$$

By solving the eigenvalue problem, the optimal solution of the projection direction w_A and w_B are achieved (Eq.3). Where 0 shows zero vector relating to the proper number of zero elements.

$$\begin{bmatrix} 0 & C_{AB} \\ C_{AB} & 0 \end{bmatrix} \begin{bmatrix} w_A \\ w_B \end{bmatrix} = \lambda \begin{bmatrix} C_{AA} & 0 \\ 0 & C_{BB} \end{bmatrix} \begin{bmatrix} w_A \\ w_B \end{bmatrix} \quad (3)$$

Kernel Canonical Correlation Analysis (KCCA) is considered as an extension of CCA which tries to maximally correlate nonlinear relationships between two various feature sets [23]. The desired projection vectors $w_A^{\emptyset}$ and $w_B^{\emptyset}$ are represented as a linear combination of all training samples existing in each feature set and there are coefficient vectors as $a = [a^1, \ldots, a^n]^T$ and $b = [b^1, \ldots, b^n]^T$. By substituting $w_A^{\emptyset} = \sum_{i=0}^{n} a^i \emptyset_A(A_i) = \emptyset(X)\, a$ and $w_B^{\emptyset} = \sum_{i=0}^{n} b^i \emptyset_B(B_i) = \emptyset(Y)\, b$ into (Eq.2) and considering the definition of kernel matrix, the following optimization (Eq.4) can be obtained for KCCA which is solved in a similar way as CCA.

$$\begin{aligned} &max_{a,b} = a^T K_A K_B b \\ &\text{Such that } a^T K_A K_A a = 1, \quad b^T K_B K_B b = 1 \end{aligned} \quad (4)$$

2) SVM-2K ALGORITHM

While SVM is a 1-dimensional projection followed by thresholding, SVM-2K integrates the two steps by establishing the constraint of similarity between two 1-dimensional projections distinguishing two well-defined SVMS [24]. In fact, SVM-2k trains SVM separately on both existing views and regularize them using constraints of similarity by a ϵ-intensive term (Eq.5)

$$\left|(w_A, \phi_A(x_i)) + b_A - (w_B, \phi_B(x_i)) - b_B\right| \leq \eta_i + \epsilon \quad (5)$$

where w_A and w_B are weight and b_A and b_B are the threshold of the first and second views respectively. Moreover, ϕ_A and ϕ_B demonstrate two feature functions while x_i is the input and η_i is the slack variable.

By combining this constraint with the common constraint of 1-norm SVM and by applying various regularization constraint, the following optimization (Eq.6) can be obtained for the classifier parameters of (w_A, b_A) and (w_B, b_B).

$$\begin{aligned}
\min L = {} & \frac{1}{2}\|w_A\|^2 + \|w_B\|^2 + C_1 \sum_{i=1}^{l} q_i^A \\
& + C_2 \sum_{i=1}^{l} q_i^B + D \sum_{i=1}^{l} \eta_i \\
\text{Such that } & \left|(w_A, \phi_A(x_i)) + b_A - (w_B, \phi_B(x_i)) - b_B\right| \\
& \leq \eta_i + \epsilon, \\
& y_i((w_A, \phi_A(x_i)) + b_A) \geq 1 - q_i^A, \\
& y_i((w_A, \phi_A(x_i)) + b_A) \geq 1 - q_i^B, \\
& q_i^A \geq 0, q_i^B \geq 0, \eta_i \geq \\
& 0 \text{ all for } (1 \leq i \leq l)
\end{aligned} \quad (6)$$

It must be taken into consideration that C_1, C_2, D and ϵ are nonnegative parameters while q_{Ai} and q_{Bi} are slack vectors. Assuming that $\acute{w_A}, \acute{w_B}, \acute{b_A}$, and $\acute{b_B}$ are the solution of this optimization problem, SVM-2K decision function can be computed as follows (Eq.7)

$$\begin{aligned}
f(x) &= \frac{1}{2}((\acute{w}_A, \phi_A(x_i)) + \acute{b}_A + (\acute{w}_B, \phi_B(x_i)) + \acute{b}_B)) \\
&= \frac{1}{2}(f_A(x) + f_B(x))
\end{aligned} \quad (7)$$

Dual problem (Eq.8) of the mentioned optimization problem can be obtained by applying normal Lagrange multipliers techniques while $\alpha_i^A, \alpha_i^B, \beta_i^+$ and β_i^- are their corresponding vectors.

$$\begin{aligned}
\max\ W = {} & -\frac{1}{2}\sum_{i,j=1}^{l} (g_i^A g_j^A k_A(x_i, x_j) + g_i^B g_j^B k_B(x_i, x_j)) \\
& + \sum_{i=1}^{l} (\alpha_i^A + \alpha_i^B) \\
\text{Such that } & g_i^A = \alpha_i^A y_i - \beta_i^+ + \beta_i^-, \\
& g_i^B = \alpha_i^B y_i - \beta_i^+ + \beta_i^-, \\
& \sum_{i=1}^{l} g_i^A = 0 = \sum_{i=1}^{l} g_i^B, \\
& 0 \leq \alpha_i^{A/B} \leq C_{1/2}, \\
& 0 \leq \beta_i^{+/-}, \quad \beta_i^+ + \beta_i^- \leq D
\end{aligned} \quad (8)$$

By considering $\epsilon = 0$, the prediction function for each view is calculated as (Eq.9):

$$fA/B(x) = \sum_{i=1}^{l} g_i^{A/B} k_{A/B}(x_i, x) + b_{A/B} \quad (9)$$

3) MVMED ALGORITHM

Multi-View Maximum Entropy Discrimination (MVMED) is considered as an extension of MED with multiple feature sets where the classification margin of each view achieved from the mentioned feature sets must be identical [27]. It means that the classification confidences from various views are supposed to match each other entirely. Considering that we are given a multi-view dataset as $\left\{X_t^1, X_t^2, y_t | 1 \leq t \leq n\right\}$, where X_t^1 and X_t^2 demonstrate the t^{th} samples from the first and second view respectively and y_t denotes their corresponding labels. MVMED tries to investigate the joint distribution over the first and second view classifier parameters ($\boldsymbol{\Theta_1}$ and $\boldsymbol{\Theta}_2$). In other words, MVMED employs joint distribution $p(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon})$ where $\boldsymbol{\Theta_1} = \{\theta_1, b_1\}, \boldsymbol{\Theta_2} = \{\theta_2, b_2\}$ and $\boldsymbol{\Upsilon} = \{\Upsilon_1, \ldots, \Upsilon_n\}$ is the common margin vector. Accordingly, the optimization problem of MVMED is calculated as follows (Eq.10):

$$\begin{aligned}
& min_{(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon})} KL(p(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon}) || p_0(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon})) \\
& \text{Such that } \int p(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon}) [y_t L_1(X_t^1 | \boldsymbol{\Theta_1}) \\
& - y_t] d\boldsymbol{\Theta_1} d\boldsymbol{\Theta_2} d\boldsymbol{\Upsilon} \geq 0 \\
& \int p(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon}) [y_t L_2(X_t^2 | \boldsymbol{\Theta_2}) \\
& - y_t] d\boldsymbol{\Theta_1} d\boldsymbol{\Theta_2} d\boldsymbol{\Upsilon} \geq 0 \\
& 1 \leq t \leq n
\end{aligned} \quad (10)$$

where $L_1(X_t^1 | \boldsymbol{\Theta_1})$ and $L_2(X_t^2 | \boldsymbol{\Theta_2})$ denote the discrimination function of the first and second views respectively. In the following, the expected large margin constraints are applied to both views. The solution of MVMED optimization problem depends on Theorem 1.

Theorem 1: *The solution to* MVMED *problem has the following general form (Eq.11):*

$$\begin{aligned}
& p(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon}) \\
& = \frac{1}{Z(\lambda_1, \lambda_2)} p_0(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon}) \\
& \times e^{(\sum_{t=1}^{n} \lambda_{1t}[y_t L_1(X_t^1 | \boldsymbol{\Theta_1}) - y_t] + \sum_{t=1}^{n} \lambda_{2t}[y_t L_2(X_t^2 | \boldsymbol{\Theta_2}) - y_t])}
\end{aligned} \quad (11)$$

where $z(\lambda_1, \lambda_2)$ *is the normalization constant and* $\lambda_1 = \{\lambda_{11}, \ldots, \lambda_{1n}\}$ *and* $\lambda_2 = \{\lambda_{21}, \ldots, \lambda_{2n}\}$ *define two sets of nonnegative Lagrange multipliers, one for each classification constraint.* λ_1 *and* λ_2 *are set by finding the unique maximum of the following jointly concave objective function (Eq.12):*

$$J(\lambda_1, \lambda_2) = -logZ(\lambda_1, \lambda_2) \quad (12)$$

After gaining λ_1 and λ_2, the distribution $p(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}, \boldsymbol{\Upsilon})$ is correspondingly characterized. In the following, as $\boldsymbol{\Upsilon}$ is marginalized, the distribution $p(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2})$ is therefore

obtained which can be employed for predicting the label of a new input sample (X^1, X^2) using the following equations (Eq.13 and Eq.14):

$$\hat{y_1} = sign(\int p(\mathbf{\Theta_1}, \mathbf{\Theta_2}) L_1(X^1|\mathbf{\Theta_1}) d\mathbf{\Theta_1} d\mathbf{\Theta_2} \quad (13)$$

$$\hat{y_2} = sign(\int p(\mathbf{\Theta_1}, \mathbf{\Theta_2}) L_2(X^2|\mathbf{\Theta_2}) d\mathbf{\Theta_1} d\mathbf{\Theta_2} \quad (14)$$

By integrating the two views, the overall prediction rule is (Eq.15):

$$\hat{y} = sign(\frac{1}{2} \int p(\mathbf{\Theta_1}, \mathbf{\Theta_2})(L_1\left(X^1 \mid \mathbf{\Theta_1}\right) + L_2\left(X^2 \mid \mathbf{\Theta_2}\right)) \times d\mathbf{\Theta_1} d\mathbf{\Theta_2}) \quad (15)$$

4) SMVMED ALGORITHM

Soft margin consistency based Multi-View Maximum Entropy Discrimination (SMVMED) is an extension of MVMED which tries to obtain margin consistency in a less strict way [28]. As a result, the relative entropy between the fundaments of two view margins is maximized. In other words, despite MVMED which employs hard margin consistency rule that forces the views to have the same margin, SMVMED attains soft margin consistency. Therefore, SMVMED is more flexible due to the profiting from a trade-off parameter balancing the large margin and margin consistency which is obtained by minimizing the KL-divergence between the fundaments of margin parameters form the two views.

Similar to MVMED, suppose that there is a multi-view dataset as $\left\{X_t^1, X_t^2, y_t | 1 \leq t \leq n\right\}$, where X_t^1 and X_t^2 demonstrate the t^{th} samples from the first and second views respectively and y_t denotes their corresponding labels. The goal of SMVMED is to learn two discriminant functions $L_1(X_t^1|\mathbf{\Theta_1})$ and $L_2(X_t^2|\mathbf{\Theta_2})$ corresponding to the first and second views respectively while $\mathbf{\Theta_1}$ and $\mathbf{\Theta_2}$ are their classifier parameters. Unlike MVMED that uses augmented joint distribution, SMVMED assumes that there are dependent distributions as $p(\mathbf{\Theta_1})$ and $p(\mathbf{\Upsilon})$ for the first view and $q(\mathbf{\Theta_2})$ and $q(\mathbf{\Upsilon})$ for the second view. Therefore, their joint distributions are $p(\mathbf{\Theta_1}, \mathbf{\Upsilon}) = p(\mathbf{\Theta_1}) p(\mathbf{\Upsilon})$ and $q(\mathbf{\Theta_2}, \mathbf{\Upsilon}) = q(\mathbf{\Theta_2}) q(\mathbf{\Upsilon})$ where $\mathbf{\Upsilon} = \{\Upsilon_t | 1 \leq t \leq n\}$ is the margin parameter. Moreover, $p(\mathbf{\Theta_1})$ and $q(\mathbf{\Theta_2})$ are respectively the posteriors of $\mathbf{\Theta_1}$ and $\mathbf{\Theta_2}$ and $p(\mathbf{\Upsilon})$ and $q(\mathbf{\Upsilon})$ are also respectively the posteriors of the margins from the first and second views. Accordingly, the optimization problem of MVMED is calculated as follows where the parameter α plays the trade-off role aiming to balance the large margin and soft margin consistency (Eq.16):

$$\begin{aligned} &min_{p(\mathbf{\Theta_1}, \mathbf{\Upsilon}), q(\mathbf{\Theta_2}, \mathbf{\Upsilon})} KL(p(\mathbf{\Theta_1}) || p_0(\mathbf{\Theta_1})) \\ &+KL(q(\mathbf{\Theta_2}) || q_0(\mathbf{\Theta_2})) \\ &+(1-\alpha) KL(p(\mathbf{\Upsilon}) || p_0(\mathbf{\Upsilon})) \\ &+(1-\alpha) KL(q(\mathbf{\Upsilon}) || q_0(\mathbf{\Upsilon})) \\ &+\alpha KL(p(\mathbf{\Upsilon}) || q(\mathbf{\Upsilon})) + \alpha KL(q(\mathbf{\Upsilon}) || p(\mathbf{\Upsilon})) \end{aligned}$$

Such that

$$\begin{aligned} &\int p(\mathbf{\Theta_1}, \mathbf{\Upsilon}) [y_t L_1(X_t^1|\mathbf{\Theta_1}) - y_t] d\mathbf{\Theta_1} d\mathbf{\Upsilon} \geq \mathbf{0} \\ &\int q(\mathbf{\Theta_2}, \mathbf{\Upsilon}) [y_t L_2(X_t^2|\mathbf{\Theta_2}) - y_t] d\mathbf{\Theta_2} d\mathbf{\Upsilon} \geq \mathbf{0} \\ &1 \leq t \leq n \end{aligned} \quad (16)$$

While finding the solution of making the partial derivatives of Lagrangian of (Eq.16) is tricky ($p(\mathbf{\Theta_1}, \mathbf{\Upsilon}) = 0$ and $q(\mathbf{\Theta_2}, \mathbf{\Upsilon}) = 0$), an iterative scheme is used for finding the solution where $p^m(\mathbf{\Theta_1}, \mathbf{\Upsilon})$ and $q^m(\mathbf{\Theta_2}, \mathbf{\Upsilon})$ are updated by solving the following two problems (Eq.17 and Eq.18):

$$\begin{aligned} &p^m(\mathbf{\Theta_1}, \mathbf{\Upsilon}) \\ &= min_{p^m(\mathbf{\Theta_1}, \mathbf{\Upsilon})} KL(p^m(\mathbf{\Theta_1}) || p_0(\mathbf{\Theta_1})) \\ &\quad +(1-\alpha) KL(p^m(\mathbf{\Upsilon}) || p_0(\mathbf{\Upsilon})) \\ &\quad +\alpha KL(p^m(\mathbf{\Upsilon}) || q(\mathbf{\Upsilon})) \end{aligned}$$

Such that

$$\begin{aligned} &\int p^m(\mathbf{\Theta_1}, \mathbf{\Upsilon}) [y_t L_1(X_t^1|\mathbf{\Theta_1}) \\ &-y_t] d\mathbf{\Theta_1} d\mathbf{\Upsilon} \geq \mathbf{0} \\ &1 \leq t \leq n \end{aligned} \quad (17)$$

And

$$\begin{aligned} &q^m(\mathbf{\Theta_2}, \mathbf{\Upsilon}) \\ &= min_{q^m(\mathbf{\Theta_2}, \mathbf{\Upsilon})} KL(q^m(\mathbf{\Theta_2}) || q_0(\mathbf{\Theta_2})) \\ &\quad +(1-\alpha) KL(q^m(\mathbf{\Upsilon}) || q_0(\mathbf{\Upsilon})) + \alpha KL(q^m(\mathbf{\Upsilon}) || q(\mathbf{\Upsilon})) \end{aligned}$$

Such that

$$\begin{aligned} &\int q^m(\mathbf{\Theta_2}, \mathbf{\Upsilon}) [y_t L_2(X_t^2|\mathbf{\Theta_2}) \\ &-y_t] d\mathbf{\Theta_2} d\mathbf{\Upsilon} \geq \mathbf{0} \\ &1 \leq t \leq n \end{aligned} \quad (18)$$

Accordingly, the solution of SMVMED optimization problem depends on the Theorem 2.

Theorem 2: *The solution to* SMVMED *problem has the following general forms (Eq.19, Eq.20):*

$$\begin{aligned} &p^m(\mathbf{\Theta_1}, \mathbf{\Upsilon}) \\ &= \frac{1}{Z_1^m(\lambda_1^m)} p_0(\mathbf{\Theta_1}) [p_0(\mathbf{\Upsilon})]^{1-\alpha} [q^{m-1}(\mathbf{\Upsilon})]^{\alpha} \\ &\quad \times e^{(\sum_{t=1}^n \lambda_{1,t}^m [y_t L_1(X_t^1 | \mathbf{\Theta_1}) - y_t])} \end{aligned} \quad (19)$$

$$\begin{aligned} &q^m(\mathbf{\Theta_2}, \mathbf{\Upsilon}) \\ &= \frac{1}{Z_2^m(\lambda_2^m)} q_0(\mathbf{\Theta_2}) [q_0(\mathbf{\Upsilon})]^{1-\alpha} [p^m(\mathbf{\Upsilon})]^{\alpha} \\ &\quad \times e^{(\sum_{t=1}^n \lambda_{2,t}^m [y_t L_2(X_t^2 | \mathbf{\Theta_2}) - y_t])} \end{aligned} \quad (20)$$

where $Z_1^m(\lambda_1^m)$ and $Z_2^m(\lambda_2^m)$ are the normalization constants and λ_1^m and λ_2^m are set by finding the maximum of the following objective functions (Eq.21, Eq.22):

$$J_1^m(\lambda_1^m) = -\log Z_1^m(\lambda_1^m) \quad (21)$$

$$J_2^m(\lambda_2^m) = -\log Z_2^m(\lambda_2^m) \quad (22)$$

After each iteration, the relative error between values of (Eq.21) and values (Eq.22) from two successful iterations are respectively calculated for determining the convergence. When the relative errors computed using the following equations (Eq.23, Eq.24) are both than less than the tolerance ϵ, the iteration ends.

$$\frac{J_1^m(\lambda_1^m) - J_1^{m-1}(\lambda_1^{m-1})}{J_1^{m-1}(\lambda_1^{m-1})} \tag{23}$$

$$\frac{J_2^m(\lambda_2^m) - J_2^{m-1}(\lambda_2^{m-1})}{J_2^{m-1}(\lambda_2^{m-1})} \tag{24}$$

After obtaining $p(\mathbf{\Theta_1})$ and $q(\mathbf{\Theta_2})$, the label of a new input sample (X^1, X^2) can be predicted using the following equations (Eq.25 and Eq.26):

$$\hat{y_1} = sign(\int p(\mathbf{\Theta_1}) L_1(X^1|\mathbf{\Theta_1}) d\mathbf{\Theta_1} \tag{25}$$

$$\hat{y_2} = sign(\int p(\mathbf{\Theta_2}) L_2(X^2|\mathbf{\Theta_2}) d\mathbf{\Theta_2} \tag{26}$$

By integrating the two views, the overall prediction rule is (Eq.27):

$$\hat{y} = sign(\frac{1}{2} \int p(\mathbf{\Theta_1}) L_1\left(X^1 \mid \mathbf{\Theta_1}\right) d\mathbf{\Theta_1} + \frac{1}{2} \int p(\mathbf{\Theta_2}) L_2\left(X^2 \mid \mathbf{\Theta_2}\right) d\mathbf{\Theta_2}) \tag{27}$$

V. EXPERIMENTS

A. EXPERIMENT DESCRIPTION

Deep neural networks have the potential to automatically extract features without human intervention. We hypothesized that the intermediate representations extracted from deep neural networks can be used as suitable features for the task of sentiment classification. On the other hand, while various deep neural networks have different structures, they are therefore able to extract different types of intermediate representations. In this regard, we decided to combine deep features extracted from convolutional and recursive neural networks to see if they are complementary or merging them with a multi-view classifier can enhance the overall efficiency.

For this aim, we employed four of the most effective multi-view classifiers in our proposed model and various experiments considering the different variations were carried out to illustrate that applying a multi-view classifier on heterogeneous single-view data can not only take advantages of their potentials but also improve the overall performance. Different variations of the proposed model are as follows. Notably, they all employ pre-trained word vectors obtained from Skip-Gram model as input.

- CNN-RNN+KCCA: Intermediate representations extracted from convolutional and recursive neural networks are fed to KCCA algorithm for classification.
- CNN-RNN+SVM2K: Intermediate representations extracted from convolutional and recursive neural networks are fed to SVM2K algorithm for classification.
- CNN-RNN+MVMED: Intermediate representations extracted from convolutional and recursive neural networks are fed to MVMED algorithm for classification.
- CNN-RNN+SMVMED: Intermediate representations extracted from convolutional and recursive neural networks are fed to SMVMED algorithm for classification.

To illustrate that applying a multi-view classifier on a single-view data obtained from convolutional and recursive neural networks can improve the classification accuracy, the proposed model must be compared to some baselines. In this regard, the obtained results are compared with some of the state-of-the-art in the family of convolutional and recursive neural networks that are explained in the following:

- CNN Basic (1-layer): Basic convolutional neural network with max-pooling [32].
- CNN-non-static: Convolutional neural network which employs word vectors that are fine-tuned for each specific task [30].
- CNN-multichannel: A convolutional neural network that employs two sets of word vectors. Each set is considered as a separate channel. Although filters are applied in both channels while gradients are propagated only through one channel [30].
- DCNN: Convolutional neural network with dynamic max-pooling [32].
- RecRNN: Basic Recursive Neural Network [50].
- RNTN: Recursive Neural Tensor Network [50].
- MVRNN: Matrix-vector recursive neural network using a parse tree [35].
- RN3: Recursive Nested Neural Network [53].

B. DATASET

All variations of the proposed model are evaluated on two different sets of Stanford Sentiment Treebank as SST1 and SST2. SST1 is the extended version of MR [54] dataset that has *train/dev/test* split and fine-grained labels. Reviews in this dataset are categorized into five categories as negative, somewhat negative, neutral, somewhat positive, and positive. SST2 is the modified version of SST1 that only includes binary labels (negative and positive) and neutral labels are eliminated [50]. It must be taken into consideration that both of these datasets are for sentence-level classification and Standard *train/test* sets of SST1, SST2 were used for conducting the experiments. Summary statistics of these two datasets after tokenization is presented in Table 1.

C. MODEL CONFIGURATION

It is worth mentioning that hyper-parameters have a great impact on the classification performance and their various combinations may lead to similar results. To obtain the optimal values for the hyper-parameters of the proposed model, an extensive range of experiments were conducted and the employed hyper-parameters are divided into two groups. The first group is related to the training of the word vectors using Skip-Gram model where the dimensionality of 150 words and

TABLE 1. Statistics of SST1 and SST2 datasets.

Dataset	Class	Average sentence length	Set / Number of sentences	
SST1	5	18	*Train*	8544
			Dev	1101
			Test	2210
SST2	2	19	*Train*	6920
			Dev	872
			Test	1821

a window size of 3 were assumed and word vectors were also updated with a learning rate of 0.025. The second group is related to the hyper-parameters of the used deep neural networks. For the convolutional part, the number of filters and filter size were treated as hyper-parameters while the hidden state dimension was considered as a hyper-parameter for the recursive part. Dropout was also used along the training to reduce overfitting.

Based on our observations, filter size (3,4,5) and 150 filters yielded to the best results. The highest performance of the recursive part was also obtained when the hidden state dimension was between 30 and 40. To reduce overfitting, the proposed model was regularized with a dropout rate of 0.5. Notably, ADADELTA update rule was used for stochastic gradient descend while a learning rate of 0.01 was used for training. Mini-bath size was equal to 25 and 50 epochs were used to train the model.

Moreover, in order to investigate the sensitivity of the proposed model to various values of hyper-parameters, we decided to investigate the effect of various hyper-parameters setting. In this regard, we hold all setting constant and vary only one factor to examine the sensitivity of the proposed model. We report the effect of the filter size, number of filters, hidden state dimension, and dropout rate on one of the variations of the proposed model (CNN-RNN+SMVMED).

- In order to investigate the effect of the filter size, various numbers of filter sizes were explored while the other parameters were kept constant. According to previous studies that demonstrated the priority of multiple filter sizes in comparison to the single filter size, we also used multiple region sizes in our experiments As can be seen in Table 2, various filter size has a great impact on the performance of the model and the greatest accuracy is obtained while the multiple filter size was set as (3, 4, 5).
- To explore the impact of the number of filters, all factors were held constant and we only changed the number of filters in each region. Based on the obtained results (Table 3), it is clear the number of filters has also considerable impact on the performance of the proposed model and the highest accuracy was obtained while the number of filters was set to 150.
- To explore the influence of the hidden state dimension, various dimensions of hidden states were explored while the other parameters were kept constant. Based on the obtained results (Table 4), the highest accuracy was obtained when the hidden state dimension was between 30 and 40.
- In order to investigate the effect of dropout as a regularization technique, different dropout rates in the range of 0.1 to 0.9 were used to find the optimal rate. Based on the obtained results (Table 5), the highest accuracy was obtained when the dropout rate was around 0.5.

TABLE 2. The effect of the filter size on the performance of the proposed model (CNN-RNN+ SMVMED).

Filter size	Accuracy %	
	SST1	SST2
(3,4,5)	**52.94**	**91.93**
(4,5,6)	52.34	91.32
(6,7,8)	52.07	91.21
(8,9,10)	52.11	91.05
(9,10,11)	51.89	90.64
(3,4,5,6)	50.90	90.02
(7,7,7,7)	51.03	90.12

TABLE 3. The effect of the number of filters on the performance of the proposed model (CNN-RNN+ SMVMED).

Number of filters	Accuracy %	
	SST1	SST2
125	52.11	91.23
150	**52.94**	**91.93**
256	51.03	90.85
300	50.98	90.35
512	51.32	89.63

TABLE 4. The effect of the hidden state dimension on the performance of the proposed model (CNN-RNN+ SMVMED).

Hidden State Dimension	Accuracy %	
	SST1	SST2
5-10	43.12	82.65
10-20	44.8	83.63
20-30	50.35	89.93
30-40	**52.94**	**91.93**
40-50	51.63	90.01
50-60	48.31	88.93
60-70	49.34	87.63

D. RESULTS AND DISCUSSION

In order to provide a fair comparison between the proposed model and other existing traditional models, a wide range of experiments were conducted and all variations of the proposed model were compared to a wide range of single-view models. The empirical results are presented in Table 6 which contains two sections. The first section of the table contains single-view models in the family of convolutional and recursive neural networks while all variations of the proposed model that occupy multi-view classifiers are depicted in the

TABLE 5. The effect of the dropout rate on the performance of the proposed model (CNN-RNN+ SMVMED).

Dropout rate	Accuracy %	
	SST1	SST2
0.1	50.38	89.93
0.2	51.31	90.12
0.3	51.78	90.92
0.4	52.03	91.43
0.5	**52.94**	**91.93**
0.6	52.63	91.21
0.7	51.34	90.89
0.8	50.87	90.63
0.9	50.04	90.34

TABLE 6. Percentage accuracies of all variations of the proposed model against other single-view models.

	Model	Dataset	
		SST1	SST2
Single-View	CNN-1 layer [32]	37.4	77.1
	CNN-non static [30]	48	87.2
	CNN-multichannel [30]	47.4	88.1
	DCNN [32]	48.5	86.8
	RecRNN [50]	43.2	82.4
	RNTN [50]	45.7	85.4
	MVRNN [35]	44.4	82.9
	RN3 [53]	47.2	81.3
Multi-View	CNN-RNN+KCCA	49.42	88.73
	CNN-RNN+SVM2K	48.3	87.45
	CNN-RNN+MVMED	52.09	90.32
	CNN-RNN+SMVMED	**52.94**	**91.93**

second section. Notably, the results of single-view models are taken from their original papers.

As it is illustrated, all variations of the proposed model perform relatively better in comparison to the other baselines. Accordingly, it can be concluded that employing multi-view learning can enhance the accuracy and therefore multi-view classifiers have considerably greater performance compared to single-view classifiers. Specifically, using SMVMED as a multi-view classifier led to the significant improvement and CNN-RNN+SMVMED outperformed the best over both datasets. On the other hand, CNN-RNN+SVM2K has the lowest accuracy on both datasets which can be due to its lower generalization ability.

Other sets of experiments were also carried out to show the importance of employing pre-trained word vectors. In this regard, all variations of the proposed model were implemented on both datasets using both pre-trained and random initialized word vectors as an input. The comparison between employing pre-trained and random initialized word vectors is presented in Table 7. As it can be clearly seen, all variations of the model that employ pre-trained word vectors perform slightly better due to the word vector representation architecture that is applied with the aim of solving semantic

TABLE 7. Comparison of the accuracy of all variations of the proposed model along random initialized word vectors and pre-trained word vectors.

Model	SST1		SST2	
	Random initialized word vectors	Pre-trained word vectors	Random initialize d word vectors	Pre-trained word vectors
CNN-RNN+KCCA	47.62	49.42	86.05	88.73
CNN-RNN+SVM2K	46.54	48.3	84.35	87.45
CNN-RNN+MVMED	50.25	52.09	89.12	90.32
CNN-RNN+SMVMED	50.89	52.94	90.25	91.93

TABLE 8. Percentage accuracies of all variations of the proposed model and two basic convolutional and recursive neural networks on randomly chosen datasets.

Model	SST1			SST2		
	1000	2000	3000	1000	2000	3000
Basic CNN	31.43	36.27	37.02	71.29	76.34	76.92
Basic RNN	38.51	41.23	43.05	74.54	79.2	81.92
CNN-RNN+KCCA	40.25	43.47	46.34	80.37	83.75	85.94
CNN-RNN+SVM2K	42.34	46.02	46.65	76.53	80.25	87.34
CNN-RNN+MVMED	44.06	47.14	47.43	83.01	**88.26**	89.03
CNN-RNN+SMVMED	**44.36**	**47.33**	**48.69**	**84.43**	87.43	**89.54**

sparsity problem. In fact, it can be claimed that using vector representation on the first layer of the proposed model has a considerable effect on classification accuracy.

While the goal of the proposed model is to enhance the performance of classification on the sparse dataset, firstly, 1000, 2000, 3000 samples of each dataset were respectively selected as the training sets while the test sets remained unchanged. In these sets of experiments, two traditional convolutional and recursive neural networks were also trained on the selected training sets individually in order to specify the strength of using multi-view classifiers in the task of sentiment analysis. The four different multi-view classifiers that are used in this paper are also tested. The obtained results are illustrated in Table 8 and Figure 6.

According to the results, it can be seen that by increasing the size of the training set, the performance of the model is increased. Therefore, it can be concluded that the proposed model used the complementarity of heterogeneous deep features to enhance the overall performance. It is worth mentioning that CNN-RNN+SMVMED has the highest accuracy on all of the randomly chosen training sets of SST1 dataset while CNN-RNN+MVMED has the highest accuracy when 2000 samples of SST2 dataset are chosen as a training set and CNN-RNN+SMVMED achieved the highest accuracy on the other chosen training sets.

Other sets of experiments were also conducted to show the amount of improvement and measure the ability of the proposed model in improving the overall performance. In this regard, the percentage improvement of the best single-view

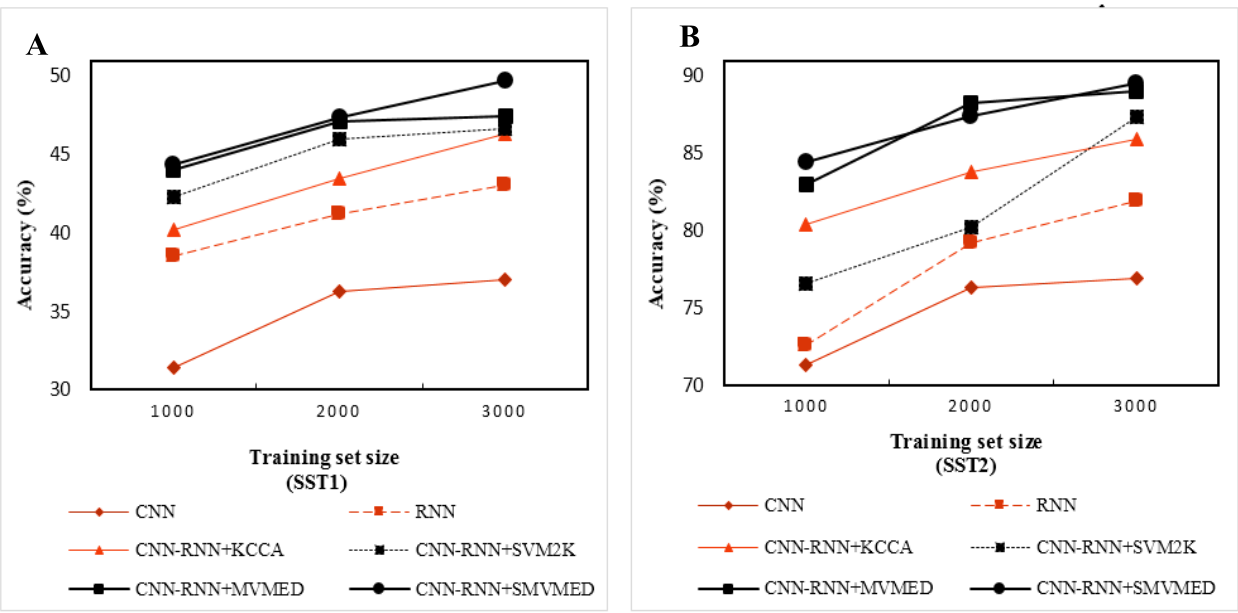

FIGURE 6. Percentage accuracies of all variations of the proposed model and two basic convolutional and recursive neural networks on randomly chosen datasets. A) SST1, B) SST2.

TABLE 9. Amount of accuracy improvement from the best single-view to the best multi-view model.

	1000	2000	3000	Average
SST1	15.21	14.8	13.11	14.37
SST2	13.27	11.45	9.31	11.34
Average	14.24	13.12	11.21	

model and the best multi-view model on each randomly chosen training set (based on Table 8) were compared and the obtained results are presented in Table 9.

According to results, it can be seen that the highest improvement is observed in the smallest training set and then by increasing the size of the training set the percentage of improvement is decreased. Therefore, it can be concluded that not only the proposed model can perform better by increasing the size of the training set but also it provides more improvement in a situation in which a small amount of data is available.

VI. CONCLUSION AND FUTURE WORK

In this paper, a new multi-view deep network for the task of sentiment analysis is proposed. Considering the analysis of the structure of various deep learning models, it can be indicated that each of them is able to extract individual intermediate representations of the input data that can be considered as a particular view. The goal of this paper is to explore the possibility of employing various features of a specific document extracted from heterogeneous neural networks by creating a multi-view framework. In this regard, intermediate features extracted from convolutional and recursive neural networks, while each of them is regarded as a single-view, are fed to four different multi-view classifiers, including SVM2k, KCCA, SVMED, and MVMED, to learn features of each view and train them jointly. It is worth mentioning that although multi-view classifiers have been extensively used in recent years, the focus of this paper is to explore the effects of using features extracted from deep neural networks which are fed to multi-view classifiers for the task of sentiment analysis.

To provide a comprehensive analysis of the performance and efficiency of the proposed model, a wide range of experiments were carried out. Accordingly, the proposed model is not only able to combine and utilize heterogeneous deep features but also outperforms the single-view deep neural networks. Moreover, it was indicated that the proposed model performs considerably better by increasing the size of the dataset as well as it provides more improvement if it is applied to a less reliable system or dataset with a fewer number of data. Generally, the proposed multi-view deep network provides a creative and intelligent framework that can apply multi-view learning on any single-view data.

Investigating future work direction contains extending the proposed model to more than two views and applying the multi-view classifier on other conventional tasks like semi-supervised learning. Moreover, other deep neural networks can be also adopted for extracting features to obtain superior performance.

REFERENCES

[1] S. A. Salloum, R. Khan, and K. Shaalan, "A survey of semantic analysis approaches," in *Proc. Int. Conf. Artif. Intell. Comput. Vis. (AICV)*. Cham, Switzerland: Springer, 2020, pp. 61–70.

[2] W. Souma, I. Vodenska, and H. Aoyama, "Enhanced news sentiment analysis using deep learning methods," *J. Comput. Social Sci.*, vol. 2, no. 1, pp. 33–46, Jan. 2019.
[3] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Inf. Fusion*, vol. 42, pp. 146–157, Jul. 2018.
[4] S. M. H. Chowdhury, S. Abujar, M. Saifuzzaman, P. Ghosh, and S. A. Hossain, "Sentiment prediction based on lexical analysis using deep learning," in *Emerging Technologies in Data Mining and Information Security*. Singapore: Springer, 2019, pp. 441–449.
[5] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, Feb. 2018.
[6] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 5, pp. 1–28, Jul. 2018.
[7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[9] H. Sadr, M. M. Pedram, and M. Teshnehlab, "A robust sentiment analysis method based on sequential combination of convolutional and recursive neural networks," *Neural Process. Lett.*, vol. 50, no. 3, pp. 2745–2761, Dec. 2019.
[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[11] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.
[12] R. Zhang, F. Nie, X. Li, and X. Wei, "Feature selection with multi-view data: A survey," *Inf. Fusion*, vol. 50, pp. 158–167, Oct. 2019.
[13] H. Sadr, M. N. Soleimandarabi, M. Pedram, and M. Teshnelab, "Unified topic-based semantic models: A study in computing the semantic relatedness of geographic terms," in *Proc. 5th Int. Conf. Web Res. (ICWR)*, Apr. 2019, pp. 134–140.
[14] Y. Li, M. Yang, and Z. Zhang, "A survey of multi-view representation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1863–1883, Oct. 2019.
[15] F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, and Y. Zhuang, "Sparse multi-modal hashing," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 427–439, Feb. 2014.
[16] S. J. Hwang and K. Grauman, "Learning the relative importance of objects from tagged images for retrieval and cross-modal search," *Int. J. Comput. Vis.*, vol. 100, no. 2, pp. 134–153, Nov. 2012.
[17] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1501–1511, Mar. 2018.
[18] Y. Jia, M. Salzmann, and T. Darrell, "Factorized latent spaces with structured sparsity," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 982–990.
[19] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, Nov. 2017.
[20] S. Sun and F. Jin, "Robust co-training," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 25, no. 7, pp. 1113–1126, Nov. 2011.
[21] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. 9th Int. Conf. Inf. Knowl. Manage. (CIKM)*, vol. 5, 2000, pp. 86–93.
[22] P. Yang and W. Gao, "Information-theoretic multi-view domain adaptation: A theoretical and empirical study," *J. Artif. Intell. Res.*, vol. 49, pp. 501–525, Mar. 2014.
[23] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.
[24] J. Farquhar, D. Hardoon, H. Meng, J. S. Shawe-Taylor, and S. Szedmak, "Two view learning: SVM-2K, theory and practice," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 355–362.
[25] S. Sun, X. Xie, and M. Yang, "Multiview uncorrelated discriminant analysis," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3272–3284, Dec. 2016.
[26] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, Jan. 2016.
[27] S. Sun and G. Chao, "Multi-view maximum entropy discrimination," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–7.
[28] L. Mao and S. Sun, "Soft margin consistency based scalable multi-view maximum entropy discrimination," in *Proc. IJCAI*, 2016, pp. 1839–1845.
[29] G. Chao and S. Sun, "Consensus and complementarity based maximum entropy discrimination for multi-view classification," *Inf. Sci.*, vols. 367–368, pp. 296–310, Nov. 2016.
[30] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882
[31] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.
[32] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*. [Online]. Available: http://arxiv.org/abs/1404.2188
[33] K. Sheng Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," 2015, *arXiv:1503.00075*. [Online]. Available: http://arxiv.org/abs/1503.00075
[34] M. Kuta, M. Morawiec, and J. Kitowski, "Sentiment analysis with tree-structured gated recurrent units," in *Proc. Int. Conf. Text, Speech, Dialogue*. Cham, Switzerland: Springer, 2017, pp. 74–82.
[35] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 1201–1211.
[36] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 689–696.
[37] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.
[38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.
[39] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078
[40] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[41] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 4, pp. 694–707, Apr. 2016.
[42] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4534–4542.
[43] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014, *arXiv:1411.2539*. [Online]. Available: http://arxiv.org/abs/1411.2539
[44] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2425–2433.
[45] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media," in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 270–278.
[46] T. Niu, S. Zhu, L. Pang, and A. El Saddik, "Sentiment analysis on multi-view social data," in *Proc. Int. Conf. Multimedia Modeling*. Cham, Switzerland: Springer, 2016, pp. 15–27.
[47] X. Wan, "Co-training for cross-lingual sentiment classification," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (ACL-IJCNLP)*, Singapore, vol. 1, 2009, pp. 235–243.
[48] P. Huang, X. Xie, and S. Sun, "Multi-view opinion mining with deep learning," *Neural Process. Lett.*, vol. 50, no. 2, pp. 1451–1463, Oct. 2019.
[49] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.; Ubiquitous Comput. Commun.; Dependable, Autonomic Secure Comput.; Pervasive Intell. Comput.*, Oct. 2015, Art. no. 23592364.
[50] R. Socher, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, A. Perelygin, and J. Wu, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2013, pp. 1631–1642.

[51] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111– 3119.
[52] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.
[53] M. Sharif and H. K. Abadi, "Recursive nested neural network for sentiment analysis," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2018.
[54] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 115–124.

HOSSEIN SADR received the Ph.D. degree in computer software engineering from Islamic Azad University, Iran, in 2020, and the M.Sc. degree in computer software engineering from Islamic Azad University, Science and Research Branch, Iran, in 2013. He is currently a Lecturer with the Department of Computer Engineering, Islamic Azad University. He is actively involved in the organization of a number of flagship conferences and workshops as well as cooperating as a Reviewer with reputable journals, such as IEEE Access and *Neural Processing Letters*. His main areas of research are natural language processing, information retrieval, machine learning, deep neural networks, and cognitive science. He is also a member of the Intelligent Systems Scientific Society of Iran (ISSSI).

MIR MOHSEN PEDRAM received the B.Sc. degree in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, 1990, and the M.Sc. and Ph.D. degrees in electrical engineering from Tarbiat Modares University, Tehran, Iran, in 1994 and 2003, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Kharazmi University. His main areas of research are intelligent systems, machine learning, data mining, and cognitive science.

MOHAMMAD TESHNEHLAB received the B.Sc. degree from Stony Brook University, USA, in 1980, the M.Sc. degree from Oita University, Japan, in 1990, and the Ph.D. degree from Saga University, Japan, in 1994. He is currently a Faculty Member of the Electrical Engineering Department, K. N. Toosi University of Technology. His research areas are artificial rough and deep neural networks, fuzzy systems and neural nets, optimization, and expert systems. He is a member of the Industrial Control Center of Excellence and the Founder of the Intelligent Systems Laboratory (ISLab). He is also a Co-Founder and a member of the Intelligent Systems Scientific Society of Iran (ISSSI) and a member of the Editorial Board of the *Iranian Journal of Fuzzy Systems* (IJFS), the *International Journal of Information and Communication Technology Research* (IJICTR), and the *Scientific Journal of Computational Intelligence in Electrical Engineering*.