IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 13, 2019, accepted December 21, 2019, date of publication January 1, 2020, date of current version January 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963426

Comparative Study of Deep Learning-Based Sentiment Classification

SEUNGWAN SEO[1], CZANGYEOB KIM[1], HAEDONG KIM[2], KYOUNGHYUN MO[3], AND PILSUNG KANG[1]

[1]School of Industrial Management Engineering, Korea University, Seoul 02841, South Korea
[2]Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, State College, PA 16801, USA
[3]SK C&C, Seoul 463844, South Korea

Corresponding author: Pilsung Kang (pilsung_kang@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grants Funded by the Korean Government (MSIT) under Grant NRF-2019R1F1A1060338 and Grant NRF-2019R1A4A1024732, and in part by the Korea Institute for Advancement of Technology (KIAT) Grant Funded by the Korean Government (MOTIE) (The Competency Development Program for Industry Specialist) under Grant P0008691.

ABSTRACT The purpose of sentiment classification is to determine whether a particular document has a positive or negative nuance. Sentiment classification is extensively used in many business domains to improve products or services by understanding the opinions of customers regarding these products. Deep learning achieves state-of-the-art results in various challenging domains. With the success of deep learning, many studies have proposed deep-learning-based sentiment classification models and achieved better performances compared with conventional machine learning models. However, one practical issue occurring in deep-learning-based sentiment classification is that the best model structure depends on the characteristics of the dataset on which the deep learning model is trained; moreover, it is manually determined based on the domain knowledge of an expert or selected from a grid search of possible candidates. Herein, we present a comparative study of different deep-learning-based sentiment classification model structures to derive meaningful implications for building sentiment classification models. Specifically, eight deep-learning models, three based on convolutional neural networks and five based on recurrent neural networks, with two types of input structures, i.e., word level and character level, are compared for 13 review datasets, and the classification performances are discussed under different perspectives.

INDEX TERMS Sentiment classification, deep learning, convolutional neural network, recurrent neural network, word embedding, character embedding.

I. INTRODUCTION

With the rapid growth of online shopping, competition has become increasingly intense as both new players and traditional offline players, such as department stores and supermarkets who have opened online stores, are constantly entering the market [1], [2]. In this e-commerce industry, millions of people express their opinions regarding purchased goods or services on popular review sites or on their personal media, such as blogs or social network services [3]. Therefore, consumers are easily affected by the online reviews of other customers when purchasing goods and services [4], [5]. Consequently, online review analysis has become a major research topic in retail businesses to discover perceptions based on review texts, which will aid in determining the appropriate managerial strategies to retain a competitive advantage against other players in the market [6]–[10]. As the volume of customer reviews is increasing significantly, it is impossible to analyze them manually; therefore, machine learning algorithms are utilized to analyze the vast amounts of customer review data [8], [11]. In machine learning, sentiment analysis generally refers to the classification of the sentiment of reviews (positive or negative); however, it generally implies the quantitative extraction of opinions, feelings, and subjectivity of texts, such as sarcasm, emoticons, and fake news detection [12], [13].

Sentiment analysis can be categorized into the following three strategies: lexicon-based, machine learning-based, and deep learning-based models. Lexicon-based models determine the sentiment of a document based on the number of

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

sentimental lexicons of each class used in the document. If more positive sentiment lexicons than negative ones exist in a document, it is classified as positive. Hence, a sentiment lexicon dictionary must be prepared prior to sentiment classification. Several previous studies [14]–[17] used existing sentiment lexicons, such as *SentiWords* [16], *MPQA lexicon* [18], and *SentiWordNet* [19], for sentiment analysis. Machine-learning-based models train a model using labeled documents. A label can be either a binary (positive or negative) or an ordinary value, such as a rating. The model performance generally depends on the employed algorithms, label correctness, and number of labeled documents. The algorithms used for sentiment analysis include naïve Bayes and support vector machine (SVM) [20]–[22]. Recently, deep-learning-based sentiment classification algorithms, especially those based on either convolutional neural networks (CNNs) or recurrent neural networks (RNNs), have demonstrated excellent performances and significantly outperformed classical lexicon-based or other machine-learning-based sentiment classification models in several studies [23]–[26].

Although deep-learning-based sentiment classification models have demonstrated outstanding performances, an optimal structure for different domains and datasets does not exist; therefore, it is difficult for practitioners when they must select an appropriate deep-learning structure for their datasets. Hence, we herein present a systematically designed empirical study comparing various deep neural network structures for sentiment analysis to investigate the relationship between classification performance and model structures/dataset characteristics. Consequently, eight structures, three of which are CNN based and the remaining five are RNN based, were considered. For each model, the word-level and character-level inputs were compared based on 13 different sentiment classification datasets. We conducted a comparative study to answer the following three primary research questions. First, how does the performance of sentiment analysis differ depending on the characteristics of the review dataset? Second, how do the basic structures, i.e., CNNs and RNNs, affect the sentiment classification performance? Third, how does the input unit, i.e., word-level and character-level inputs, affect the sentiment classification performance? Although some of the aforementioned applications, such as sarcasm or fake-news detection, are critical in understanding writer sentiment, datasets that are publicly available are insufficient for training complex structures of deep-learning-based models [27], [28]. Because polarity classification is the most fundamental but widely used task in the sentiment-analysis domain [29], we first addressed this specific task (i.e., classifying a document into either a positive or negative class) and focused on comparing the performance with respect to different deep-learning structures.

The remainder of this paper is organized as follows. In Section 2, we briefly review related studies focusing on traditional and deep-learning-based sentiment analyses. Section 3 describes the eight selected deep learning-based models, the datasets, and the experimental design. The experimental results are discussed in Section 4. Finally, we conclude our study with future studies in Section 5.

II. RELATED WORK

A. EARLY HISTORY OF SENTIMENT ANALYSIS

Sentiment analysis studies in the early days were primarily based on cognitive psychological studies that explore human intelligence quantitatively [30]; however, the majority of current studies focused on building statistical or machine-learning models based on a large labeled dataset aided by an easy access to massive review texts with ratings for various products and services through the Internet [31], [32]. For example, Nasukawa and Yi [31] proposed a sentiment classification model of online web pages based on the results of natural language processing, such as syntactic parsing and sentiment lexicon extraction. Yu and Hatzivassiloglou [32] developed a model that predicted not only the sentiment of a document, but also that of each sentence in the document for online news. Certain studies attempted to enhance the performance using handcrafted features [33] or applying various machine-learning models in online reviews [34].

B. DEEP LEARNING FOR SENTIMENT ANALYSIS

As deep-neural-network-based classification models have demonstrated significant results that outperformed conventional models in several domains, such as computer vision [35]–[38] and natural language processing [39], [40], they are currently adopted for sentiment analysis tasks. CNNs and RNNs are two typically used primary structures. CNNs assumes the form of a matrix or tensor as an input instead of a vector in a feedforward neural network. In a convolution operation, a fixed size submatrix known as a receptive field is used and produces a scalar value by adding the element-wise products between the receptive field and the convolution filter. This convolution operation is repeated from the top-left to the bottom-right of the input matrix/tensor by striding the filter. The size of the convolution, i.e., the number of convolution filters, and the stride of the convolution are hyperparameters. Pooling is another main building block of CNN that reduces the output size of the convolution layer by calculating the average or maximum value of a certain area [41]. The CNN was originally developed in the computer vision field; however, recently, it has been employed in text analytics [42]. Several recent studies have demonstrated that the CNN can learn the hierarchical structure of a language and efficiently handle variable lengths [43]–[46]. Kalchbrenner *et al.* [44] designed a network with two convolution and pooling layers and conducted an experiment with the Stanford Sentiment Treebank (SST) and Twitter sentiment datasets. Experimental results showed that the proposed CNN structure yielded better performance than regular feedforward networks and the SVM. Other studies, such as [43], achieved improved performances by adding more convolution and pooling layers and employing appropriate regularization techniques, such as dropout [47].

It has been demonstrated that the performance of CNNs can be improved when the number of layers is increased, provided that the designed architecture facilitates the gradient flow from the loss function to the input matrix/tensor [36]. However, Kim [45] showed that a shallow-and-wide CNN architecture that contains only one convolution layer and hundreds of filters can perform well for sentiment analysis; it achieved an average of 80 to 90% accuracy on online movie, SST, and product review datasets. Le *et al.* [48] conducted additional experiments and demonstrated that the shallow-and-wide structure could outperform networks with deep structures. The aforementioned studies used words as a basic unit to form the input matrix. The width of the input matrix is associated with the word vector size, while the height of input matrix is associated with the number of words in a document. In this representation, the i^{th} row corresponds to the i^{th} word in the document. Zhang *et al.* [49] presented the first study to use a character-level input on CNNs instead of a word-level input. Kim [45] classified the sentiment of a sentence using one convolution layer and one fully connected layer, while Zhang *et al.* [49] extracted text features using six convolution layers and classified a document through three fully connected layers. They used the news articles from the AG corpus,[1] Sogou News [50], DBPedia [51], Yelp Review, Yahoo! Answers dataset, and Amazon reviews [52] to evaluate the performance of the proposed model. The experimental results showed an error rate of 5% on average for the binary classification tasks, such as Yelp Review and the Amazon reviews. Conneau *et al.* [53] analyzed the sentiment of text through a CNN with a deeper layer than those used in previous studies. They employed a CNN structure with 29 layers, which was originally proposed by Simonyan and Zisserman [35], to construct a model. The experimental results based on the same datasets used by Zhang *et al.* [49] showed that the proposed model achieved an error rate of 4.5% on average for binary classification tasks.

The RNN is a deep-learning model that specializes in processing sequential data [41]. Because text is inherently sequential data, i.e., a sentence or document is a sequence of words, the RNN is typically used in text analytics. However, the recurrent structure of the RNN for processing a sequence causes challenges in learning long-term dependency in a text caused by either a vanishing or exploding gradient problem [54]. Hochreiter and Schmidhuber [55] and Cho *et al.* [56] resolved these problems by inserting a gate unit, i.e., a long short-term method (LSTM) cell or gated recurrent unit (GRU) cell. LSTM and GRU demonstrated better performances than the vanilla RNN, which has no memory cell, according to [57]; however, their performances did not differ significantly. The GRU is a special case of LSTM, i.e., it is a simplified LSTM that reduces the number of parameters for learning by combining the input and forget gates of LSTM [58]. Many studies employed either LSTM or the GRU for sentiment analysis tasks and constantly demonstrated favorable performances [59]–[62].

C. COMPARATIVE STUDIES ON SENTIMENT ANALYSIS BASED ON DEEP LEARNING

Although many studies proposed their own deep neural network structures based on the CNN or RNN for sentiment classification tasks, only a few studies have systematically compared the performances of various deep-learning-based sentiment classification models. Hu *et al.* [63] demonstrated that deep-learning-based models outperformed traditional algorithms such as dictionary-based algorithms, the SVM, or naïve Bayes, on sentiment analysis. However, they did not provide quantitative performance parameters such as F1 scores or accuracy. Yin *et al.* [64] compared the sentiment-classification performances of the CNN, LSTM, and GRU. However, their experimental results were of limited value because they did not consider sufficient model-structure variations, and their conclusion was derived from experimental results on only a single dataset. Ouyang *et al.* [43] and Singhal and Bhattacharyya [65] compared basic CNN and RNN structures, but they did not consider variations in terms of input type and model architecture. Katić and Milićević [66] evaluated CNN and LSTM performances on an Amazon review dataset. However, it was difficult for readers to derive practical guidelines because their experimental setup was not described sufficiently. Zhang *et al.* [13] presented nine studies that employed deep-learning structures, including the CNN and RNN, but performance comparisons were not provided. Hence, we conducted a systematically designed comparative experiment using eight model architectures with two input types (word and character levels) based on 13 online review datasets obtained from various domains.

III. MODELS

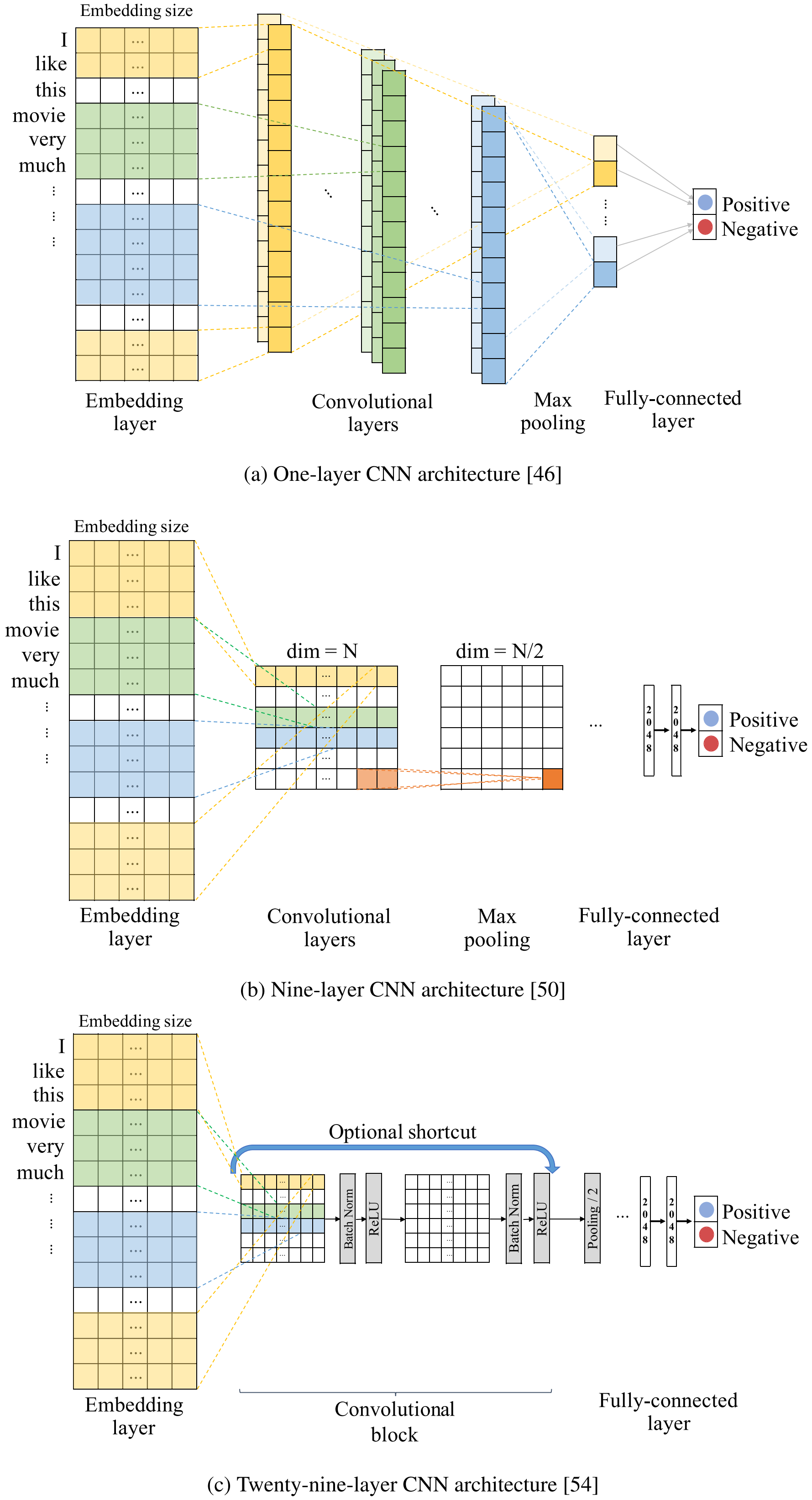
In this section, we briefly describe the eight benchmarked models: three CNN-based models and five RNN-based models.

A. CONVOLUTIONAL NEURAL NETWORK (CNN)-BASED MODELS

1) ONE-LAYER CNN

The first model that we selected was a CNN model with only one convolution layer, which was proposed in [45], as shown in Figure 1a. It uses a one-dimensional instead of a two-dimensional convolution filter for image processing. When extracting local features from images using convolution, both horizontal and vertical spatial information are important; consequently, a square convolution operation is repeated by sliding the receptive field from the left to the right and from the top to the bottom. However, as each row of the input matrix for text processing is a distributed representation of a word or character, only the vertical spatial relationship is informative. Hence, a rectangular-shaped

[1] http://www.di.unipi.it/Ëoegulli/AG_corpus_of_news_articles.html

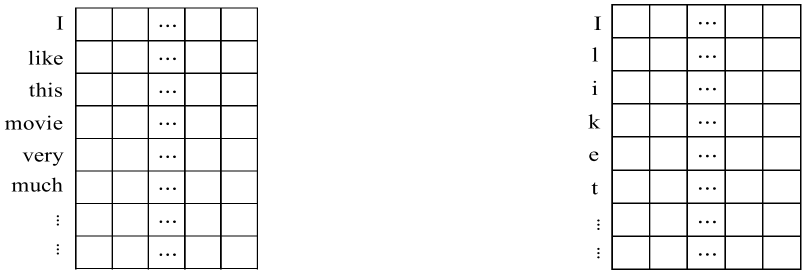


(a) One-layer CNN architecture [46]

(b) Nine-layer CNN architecture [50]

(c) Twenty-nine-layer CNN architecture [54]

FIGURE 1. **Architecture of convolutional neural network (CNN)-based models.**



(a) CNN architecture using word-level input (b) CNN architecture using character-level input

FIGURE 2. **Difference between input forms in CNN-based model.**

convolution filter whose width is the same as the width of the input matrix was used. As the width of the convolution filter and input matrix are identical, only vertical striding is necessary. In this structure, the length of an input sentence n is a fixed variable and represented as

$$x_{1:n} = x_1 \oplus x_2 \oplus \cdots \oplus x_n, \tag{1}$$

where $\oplus$ is the concatenate operation, and x_i is the i^{th} word in the sentence. If the sentence is shorter than the fixed length, then zero pads are added to the end of the input matrix. The feature c is created by a convolutional operation using consecutive h words and a filter. For example, the feature c_1 is generated as

$$c_1 = f(\mathbf{w} \cdot x_{1:h} + b), \tag{2}$$

where b is a bias term; $\mathbf{w}$ and $f(\cdot)$ represent the weight of the convolution filter and nonlinear function, respectively. The set c_i constitutes the feature map $\mathbf{c} = [c_1, c_2, \cdots, c_i]$. For each feature map, max pooling is applied to $\mathbf{c}$ to obtain the maximum value of the feature map, $\hat{c} = max\{c\}$, to extract the most important words for each convolution filter. Kim [45] used only one convolutional layer with various filter sizes. Specifically, three filter sizes, i.e., 3, 4, and 5 with 100 feature maps each, were used. Once the max pooling operation has completed, a 300-dimensional vector was generated, and it was fully connected to the output layer with two nodes: positive and negative. Between the last hidden layer and the output layer, dropout was applied to effectively regularize the model complexity. While considering the input matrix, Kim [45] used four different strategies for word vectors. In the CNN-rand model, word vectors were randomly initialized and trained together with other network parameters during training. In the CNN-non-static model, word vectors were fine-tuned after initialization using the word2vec method, while pretrained word vectors were not modified in the CNN-static model. Because the experimental results did not show a significant difference among the three models, we adopted the CNN-rand model to compare the benchmarked architectures based on an end-to-end learning scheme.

2) NINE-LAYER CNN

Zhang *et al.* [49] used a CNN architecture with six convolution layers followed by three fully connected layers for sentiment classification, which was much deeper than that used by Kim [45], as shown in Figure 1b(b). The key module in this architecture is the temporal convolution. Assuming that a discrete input function $g(x) \in [1, l]$ and a discrete kernel function $f(x) \in [1, k]$ exist, the convolution $h(y)$ between $f(x)$ and $g(x)$ is defined as

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot d - x + c), \tag{3}$$

where $c = k\ (kernel\,size) - d\ (stride) + 1$ is an offset constant. In equation 3, $f(x)$ is a weight function, and $g(x)$ and $h(y)$ are the input and output features, respectively. It eventually performed the same role as the one-dimensional convolution used by Kim [45]. They used a rectified linear unit (ReLU) as an activation function and performed max pooling between convolution layers. Additionally, they used dropout with a probability of 0.5 for the last three fully connected layers.

Another main difference between this model and that used by Kim [45] is that the input level was changed from word to character. Each character was represented by a 70-dimensional one-hot encoding vector (26 English letters, 10 digits, the new line character, and 33 other characters), and the maximum length of an input sentence was limited to 1,024 characters. Sentences with more than 1,024 characters were trimmed, while those with less than 1,024 characters were zero padded.

Although Zhang *et al.* [49] only used the character-level input for the CNN model for sentiment classification, we used both word- and character-level inputs to this model to compare the effect of the input level for the same architecture.

3) TWENTY-NINE-LAYER CNN

Conneau *et al.* [53] demonstrated that deep-learning architectures for natural language processing (NLP) were relatively shallower than those used for vision tasks. Furthermore, they discovered that many NLP approaches using words as basic units were limited in representing complex semantic information of text. Hence, Conneau *et al.* [53] proposed another deep neural network called very deep CNN, which comprised 29 layers of convolution blocks. This architecture was based on the visual geometry group network; therefore, only 3 × 3 convolution operations were used. The convolutional block comprised two convolutional layers followed by a batch normalization layer and ReLU activation. The overall model structure is shown in Figure 1c(c). The authors hypothesized that without various filter sizes, the model can learn how to best combine *"3-gram features"* through deep structures and considered both short- and long-span relations. In addition, they used the optional shortcut proposed by He *et al.* [36] to train the model more effectively. Similar to the nine-layer CNN, the input length was fixed to 1,024 such that longer sentences were trimmed and shorter sentences were zero padded.

B. RNN-BASED MODELS

Although the RNN is highly flexible for input–output mapping according to task type, e.g., many-to-many for translation or one-to-many for image captioning, we only considered many-to-one mapping for sentiment classification. Hence, each word or character was sequentially provided, and the sentiment class was determined after the final token in the sentence was provided. In addition, we compared the performances of the vanilla RNN, LSTM, and GRU to identify if different types of RNN cells affected the sentiment classification results. A bidirectional RNN method [67], which is known to improve the performance of RNNs, was also considered.

1) VANILLA RNN

The hidden state vector of the vanilla RNN first combines the current input vector, previous hidden state vector, and bias term with their corresponding weight matrices and performs a nonlinear transformation to produce the output vector, as shown in equation (4):

$$\mathbf{h}_t = tanh(\mathbf{x}_t\mathbf{U} + \mathbf{h}_{t-1}\mathbf{W} + \mathbf{b}_t). \tag{4}$$

Contrary to the CNN architectures discussed above, this recurrent structure of the RNN allows the model to preserve the sequence information. However, with the vanilla RNN architecture, this memory function does not perform well for a long sequence because of the gradient vanishing/exploding problem.

2) LONG SHORT-TERM MEMORY

To resolve the long-term dependency and vanishing/exploding gradient problem, LSTM uses the cell state to adaptively adjust the amount of historical memory and the currently provided new information [55]. LSTM comprises two state vectors: hidden state h_i and cell state C_i, and three gates: forget gate f_t, input gate i_t, and output gate o_i. Each state and gate is computed as follows:

$$\begin{aligned} f_t &= \sigma(\mathbf{W}_f \cdot [h_{t-1}, x_t] + b_f); && (5)\\ i_t &= \sigma(\mathbf{W}_i \cdot [h_{t-1}, x_t] + b_i); && (6)\\ \widetilde{C}_t &= tanh(\mathbf{W}_c \cdot [h_{t-1}, x_t] + b_c); && (7)\\ C_t &= f_t \times C_{t-1} + i_t \times \widetilde{C}_t; && (8)\\ o_t &= \sigma(\mathbf{W}_o \cdot [h_{t-1}, x_t] + b_o); && (9)\\ h_t &= o_t \times tanh(C_t). && (10) \end{aligned}$$

Both the forget gate f_t in Eq. (5) and the input gate i_t in Eq. (6) consider the previous hidden state vector h_{t-1} and the current input vector x_t and use the sigmoid as an activation function. The only difference between them are the weight matrices $\mathbf{W}_f$ and $\mathbf{W}_i$, which are learned during training. The forget gate determines the amount of previous information that should be preserved, whereas the input gate determines how much new information computed using Eq. (7) should be added when computing the current cell state, as shown in Eq. (8). Additionally, the output gate considers h_t and x_t to compute the current hidden state vector, as computed using Eq. (10). By introducing the cell state, which is represented as a red line in Figure 3, LSTM demonstrates better memorization ability for long sequences than the vanilla RNN [55], [68].

3) GATED RECURRENT UNIT

Although LSTM has proven its better performance when compared to the vanilla RNN, its computational complexity is significantly higher owing to additional weight matrices. The GRU cell was introduced to reduce the computational complexity of LSTM by combining the input and forget gate of LSTM into a single *update gate* and combining the hidden and cell states into a single hidden state, as shown in Eqs. (11)–(14) [56]. Each gate and state of the GRU was computed as follows:

$$\begin{aligned} z_t &= \sigma(\mathbf{W}_z \cdot [h_{t-1}, x_t] + b_z); && (11)\\ r_t &= \sigma(\mathbf{W}_i \cdot [h_{t-1}, x_t] + b_r); && (12)\\ \widetilde{h}_t &= tanh(\mathbf{W}_c \cdot [r_t \times h_{t-1}, x_t] + b_c); && (13)\\ h_t &= (1 - z_t) \times h_{t-1} + z_t \times \widetilde{h}_t, && (14) \end{aligned}$$

where z_t, r_t, and h_t are the update, reset, and hidden states, respectively.

4) BIDIRECTIONAL MODEL

To overcome the shortcoming of the one-directional RNN, which only considers previous context, a bidirectional RNN (BDRNN), which can consider both past and future context, was proposed by Schuster and Paliwal [67]. The BDRNN uses a concatenated forward hidden state $\overrightarrow{h}$ and backward hidden state $\overleftarrow{h}$, as shown in Eqs. (15) and (16), to produce the hidden state at time t.

$$\begin{aligned} \overrightarrow{h} &= \sigma(\mathbf{x}_t\mathbf{U} + \overrightarrow{\mathbf{h}}_{t-1}\mathbf{W} + \mathbf{b}_t) && (15)\\ \overleftarrow{h} &= \sigma(\mathbf{x}_t\mathbf{U} + \overleftarrow{\mathbf{h}}_{t+1}\mathbf{W} + \mathbf{b}_t) && (16) \end{aligned}$$

In this study, bidirectional LSTM and bidirectional GRU models were used for comparing sentiment classification performances.

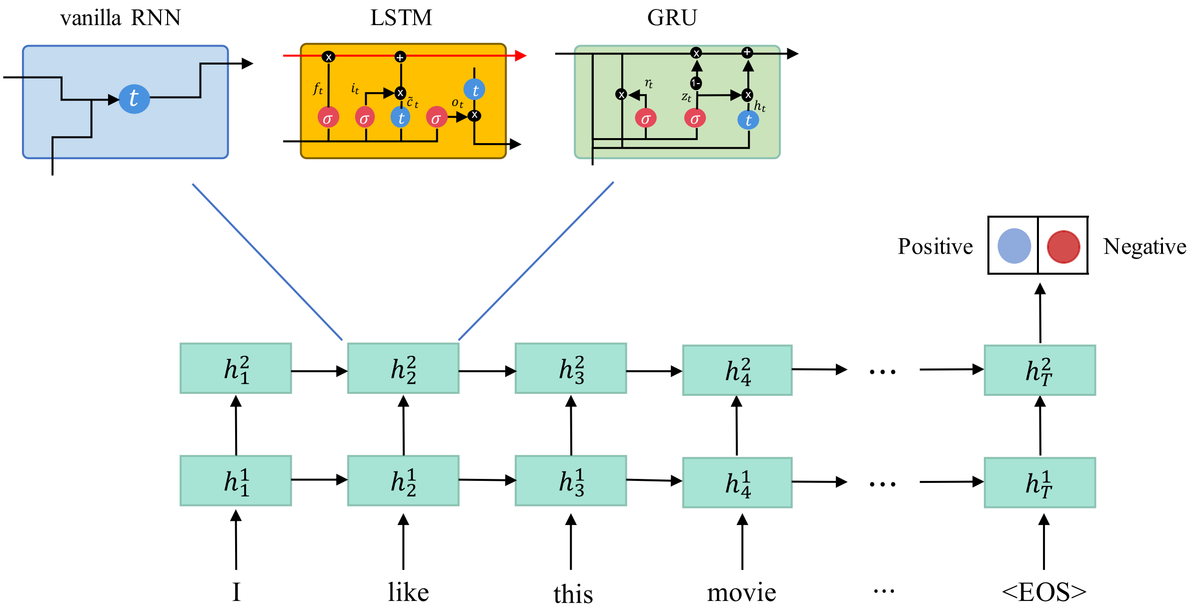
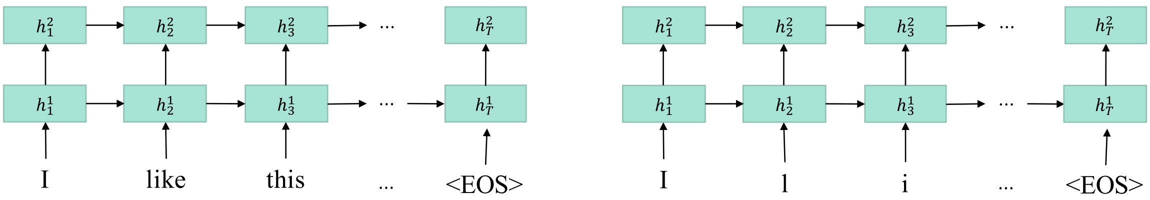


FIGURE 3. Architecture of the overall recurrent neural network (RNN)-based models and description of each cell.



(a) RNN architecture when using word-level input

(b) RNN architecture when using character-level input

FIGURE 4. Difference between input forms in the RNN-based model.

IV. EXPERIMENTS

A. DATA

To derive a generally accepted empirical conclusion, we used 13 review datasets collected for various products and services, as listed in Table 1. The first and second columns denote the expanded and shortened dataset names, respectively. All datasets were divided into training, validation, and test datasets in the proportion of 50%, 20%, and 30%, respectively. The number of reviews in the total/training/validation/test sets are listed in the third, fourth, fifth, and sixth columns of the table, respectively. The average length, vocabulary size, and positive/negative ratio are provided in the subsequent columns. The first 10 datasets are product review datasets for different categories provided by Amazon.[2] The 11th dataset is associated with Amazon's food. The 12th dataset is a hotel review dataset provided by Carnegie Mellon University, [3] and the final dataset is a movie review dataset provided by Stanford University.[4]

Those datasets were composed using five satisfaction rates (1: very negative, 5: very positive). We considered ratings 4 and 5 as positive and ratings 1 and 2 as negative. Review texts with rating 3, which is neutral, were removed from the dataset.

The datasets present different positive/negative ratios ranging from 1.06 (SST-Fine) to 13.14 (Sports and Outdoors), indicating that (1) positive reviews generally outnumber negative reviews and (2) many datasets are highly imbalanced. As model training can be difficult when the class imbalance is severe, we sampled the same number of positive and negative reviews for every mini-batch training dataset. More specifically, we set the mini-batch size to 128 with 64 positive reviews and 64 negative reviews. Consequently, individual negative reviews instead of individual positive reviews were used more frequently to train the model.

B. SENTIMENT CLASSIFICATION MODELS

In this study, we compared three CNN-based and five RNN-based models, as listed below, for sentiment classification:

[2] http://jmcauley.ucsd.edu/data/amazon
[3] http://www.cs.cmu.edu/~jiweil/html/hotel-review.html
[4] https://ai.stanford.edu/~amaas/data/sentiment

TABLE 1. Data description.

Data	Abbreviation	No. Total	No. Training	N. Validation	N. Test	Average length	Vocabulary size	Pos/Neg ratio
Tools and Home Improvement	Tools	134,476	67,239	26,895	40,342	111.86	81,250	11.24
Video Games	Video	231,780	115,890	46,356	69,534	208.66	159,820	6.13
Clothing Shoes and Jewelry	Clothing	278,677	139,339	55,735	83,603	61.21	69,887	8.31
Pet Supplies	Pet	157,836	78,919	31,567	47,350	89.8	67,558	7.03
Sports and Outdoors	Sports	296,337	148,169	59,267	88,901	88.75	98,278	13.14
Baby	Baby	160,792	80,397	32,158	48,237	100.56	59,190	7.43
Toys and Games	Toys	167,597	83,799	33,519	50,279	101.92	77,105	12.74
Grocery and Gourmet Food	Grocery	151,254	75,628	30,250	45,376	95.05	73,349	8.76
Beauty	Beauty	198,502	99,252	39,700	59,550	90.07	73,756	7.01
Cell Phones and Accessories	Phone	194,439	97,221	38,887	58,331	92.99	75,189	6.1
Amazon Fine Food Review	Food	568,454	284,228	113,690	170,536	82	75,684	5.4
TripAdvisor	Trip	878,561	439,281	175,712	263,568	145.26	219,449	5.63
SST-Fine	SST	9,613	4,806	1,923	2,884	19.3	16,173	1.06

TABLE 2. Hyperparameters for our experiments.

	Learning rate	Embedding dimension	Hidden size	Number of layers	Mini-batch size
Value	1e-3	128	128	2	128

* The hidden size and number of layers were used only for RNN-based models.

TABLE 3. Number of parameters in each model.

Data	CNN 1-layer	CNN 9-layer	CNN 29-layer	Vanilla RNN	LSTM	BD LSTM	GRU	BD GRU
No. of parameters	154,502	3,195,906	16,851,394	66,306	264,450	659,714	198,402	494,850

1) one-layer CNN: CNN with one convolutional layer [45]
2) nine-layer CNN: CNN with six convolutional layers and three dense layers [49]
3) 29-layer CNN: CNN with 26 convolutional layers and three dense layers [53]
4) Vanilla RNN: Vanilla RNN with two hidden layers
5) LSTM: LSTM cell with two hidden layers [55]
6) GRU: GRU cell with two hidden layers [56]
7) LSTM-bidirection: Bidirectional LSTM with four hidden layers
8) GRU-bidirection: Bidirectional GRU with four hidden layers

We used word- and character-level inputs for the eight models above to investigate the effect of input type on sentiment classification performance. The hyperparameter settings used in this study are listed in Table 2. According to Cui et al. [69], increasing the number of RNN layers did not guarantee a performance improvement. If the number of hidden layers is greater than two, then the computational complexity is significantly increased, while the performance is only marginally improved or even decreased occasionally. Hence, we used two hidden layers for a unidirectional RNN and four (two for forward and two for backward) hidden layers for a bidirectional RNN. Because the main purpose of our study is to investigate how structural differences between deep-learning models affect sentiment analysis, we used the same set of hyperparameters to prevent variations in performance and focus on the effects of structural differences. Table 3 lists the numbers of parameters used in each model. It is noteworthy that the parameters for the look-up table for word/character-level embedding were excluded to compare the complexities of different deep-learning models. During training, we used the Adam [70] optimizer for CNN-based models and RMSProp [71] for RNN-based models.

C. EXPERIMENTAL RESULTS

Tables 4 and 5 list the area under the receiver operating characteristics curve (AUROC) of each model with word- and character-level inputs, respectively. The last row shows the number of datasets for which each model yielded the best performance. When the word-level input was used, the RNN-based models outperformed the CNN-based models, as superior performance was reported for 11 datasets of the bidirectional LSTM, GRU, and bidirectional GRU models, while only two datasets of the CNN 1-layer model demonstrated superior performance. However, when a character-level input

TABLE 4. Area under the receiver operating characteristics curve (AUROC) of each model with word-level input.

Data	CNN 1-layer	CNN 9-layer	CNN 29-layer	Vanilla RNN	LSTM	BD LSTM	GRU	BD GRU
Tools	0.7902	0.5790	0.5897	0.7262	0.8125	0.8029	**0.8134**	0.7814
Video	0.8267	0.5785	0.7702	0.8173	0.8638	0.8849	0.8854	**0.8858**
Clothing	**0.8725**	0.6744	0.8367	0.8287	0.8576	0.8572	0.8581	0.8576
Pet	0.8215	0.5274	0.7589	0.7481	0.8527	0.8529	0.8546	**0.8351**
Sports	**0.8449**	0.6238	0.7495	0.7113	0.8085	0.8251	0.8287	0.8007
Baby	0.8534	0.5425	0.7669	0.7921	0.8770	0.8841	**0.8876**	0.8772
Toys	0.8617	0.6319	0.7378	0.8071	0.8687	0.8503	**0.8676**	0.8662
Grocery	0.8307	0.5853	0.7565	0.7813	0.8640	**0.8755**	0.8540	0.8499
Beauty	0.8549	0.5174	0.8037	0.8224	0.8874	0.8900	**0.8972**	0.8898
Phones	0.8539	0.5355	0.8041	0.8459	0.8822	**0.8909**	0.8889	0.8838
Food	0.8559	0.7663	0.8518	0.8409	0.8781	**0.8903**	0.8846	0.8890
Trip	0.9120	0.7865	0.8743	0.8720	0.9147	**0.9158**	0.9108	0.9142
SST	0.6231	0.5365	0.5009	0.6225	0.6672	**0.6863**	0.6805	0.6684
Best cases	2	-	-	-	-	**5**	4	2

TABLE 5. AUROC of each model with character-level input.

Data	CNN 1-layer	CNN 9-layer	CNN 29-layer	Vanilla RNN	LSTM	BD LSTM	GRU	BD GRU
Tools	**0.8166**	0.5409	0.6949	0.5495	0.7750	0.7788	0.7686	0.7431
Video	**0.8530**	0.5501	0.8285	0.5899	0.8346	0.8441	0.8351	0.8507
Clothing	0.8572	0.6216	**0.8599**	0.5418	0.8311	0.8338	0.8330	0.8338
Pet	0.8299	0.5938	0.8016	0.5571	0.8209	**0.8392**	0.8246	0.8078
Sports	0.8119	0.5746	**0.8143**	0.5215	0.7405	0.7719	0.7689	0.7695
Baby	0.8252	0.5714	0.8301	0.5992	0.8416	**0.8554**	0.8385	0.8100
Toys	0.8251	0.5785	0.8293	0.5904	0.8139	**0.8464**	0.8302	0.8432
Grocery	**0.8285**	0.5437	0.8029	0.6079	0.8110	0.8262	0.8216	0.8055
Beauty	0.8213	0.6141	0.8416	0.5995	0.8604	**0.8716**	0.8540	0.8369
Phones	0.8282	0.6281	0.8436	0.5939	0.8626	0.8620	**0.8631**	0.8606
Food	0.8584	0.7027	**0.8811**	0.5277	0.8405	0.8553	0.8582	0.6046
Trip	0.8759	0.6757	0.8853	0.5770	0.8824	0.8840	0.8879	**0.8912**
SST	**0.6662**	0.5308	0.6010	0.4764	0.5266	0.5250	0.5274	0.5362
Best cases	**4**	-	3	-	-	**4**	1	1

was used, it was difficult to determine the best structure as the CNN-based models yielded superior performance for seven datasets, while the RNN-based models yielded superior performance for six datasets.

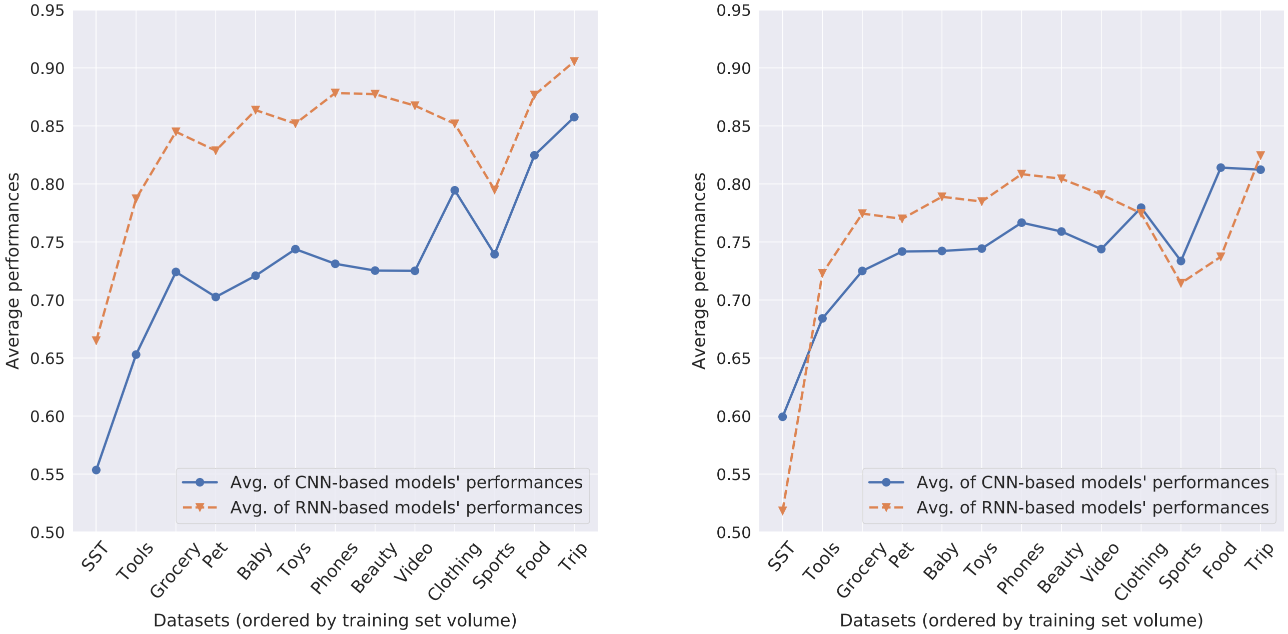
1) PERFORMANCE COMPARISON BY DATASET CHARACTERISTICS

The model performance corresponding to the training dataset volume is shown in Figure 5. The x-axis is the shortened dataset name in ascending order of the training dataset volume, and the y-axis shows the average performance of the CNN-based models as a solid blue line and the RNN-based models as a dashed orange line. The difference between Figures 5a and 5b is based on the input data level, i.e., word-level input for the former and character-level input for the latter. Generally, sentiment classification accuracy is improved when a greater number of examples are trained, irrespective of the input data level. Between the model structures, if the other conditions are identical, then the RNN-based models outperforms the CNN-based models in general for both input data levels; furthermore, the difference becomes more significant when word-level inputs are used.

The model performance for different input data levels with regard to the vocabulary size, i.e., number of unique words in the training dataset, is shown in Figure 6. In contrast to the training dataset volume, a consistent trend corresponding to the vocabulary size for both RNN- and CNN-based models does not exist, except for the SST-Fine dataset, which has a relatively small vocabulary size (16,173 words).

2) PERFORMANCE COMPARISON BY INPUT LEVEL

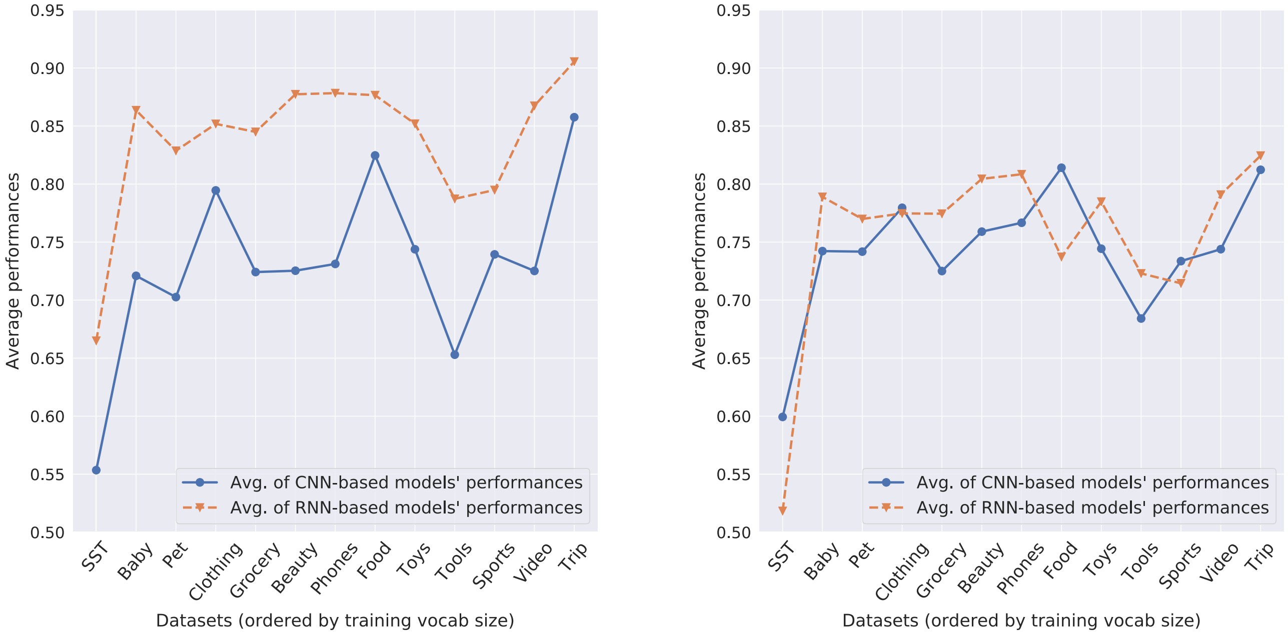
As the word-level input yielded a higher sentiment classification performance than the character-level input, we conducted an in-depth analysis to investigate the performance difference



(a) When using word-level input

(b) When using character-level input

FIGURE 5. **Average performances of CNN- and RNN-based models on each dataset (ordered according to dataset volume).**



(a) When using word-level input

(b) When using character-level input

FIGURE 6. **Average performances of CNN- and RNN-based models on each dataset (ordered according to vocabulary size).**

at the individual model level. Figure 7 shows the difference in AUROC values between the word- and character-level inputs (AUROC of word-level input − AUROC of character-level input) for each dataset. As shown in Figure 7, the input level affects the performance of the individual model structure. The CNN-based models reacted differently to the input level according to the depth of the convolution layers. With the shallow structure, the word-level input yielded a higher performance than the character-level input (all datasets except three resulted in a higher AUROC with a word-level input when compared to the character-level input for the one-layer CNN). Conversely, if the model structure becomes

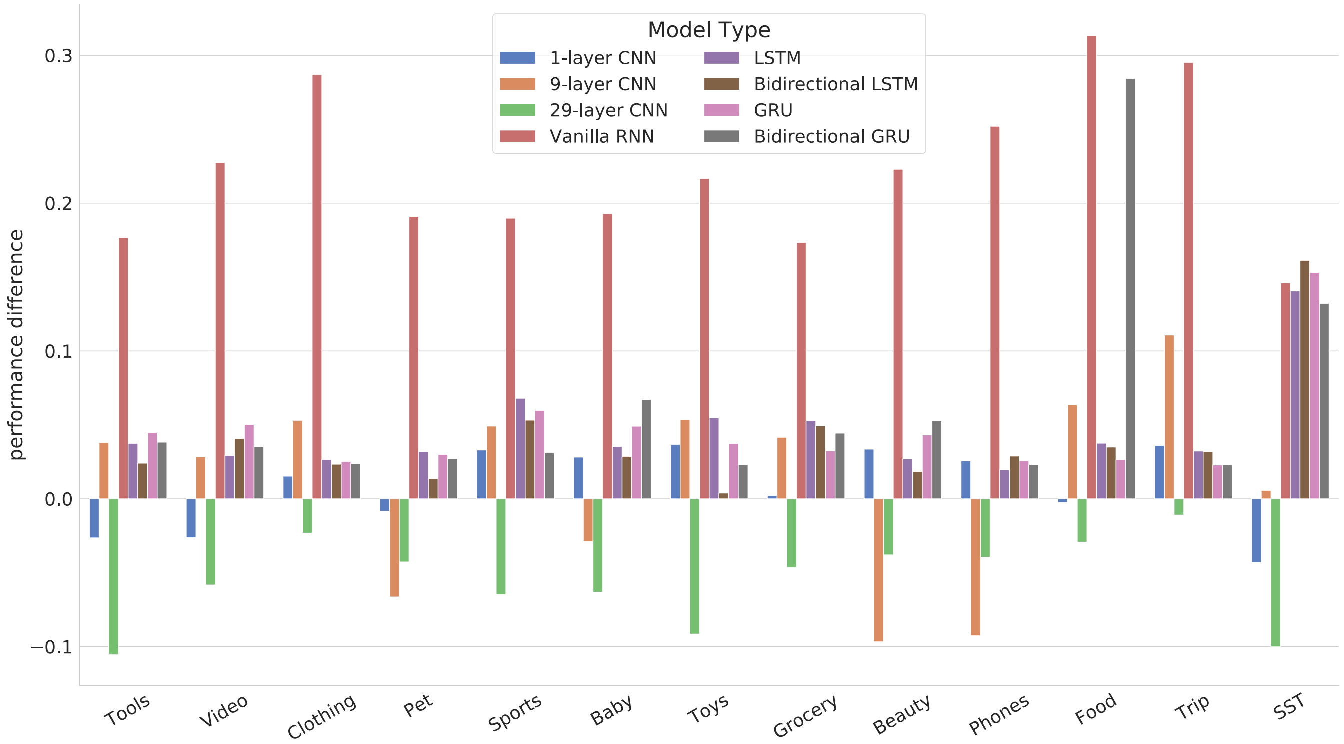


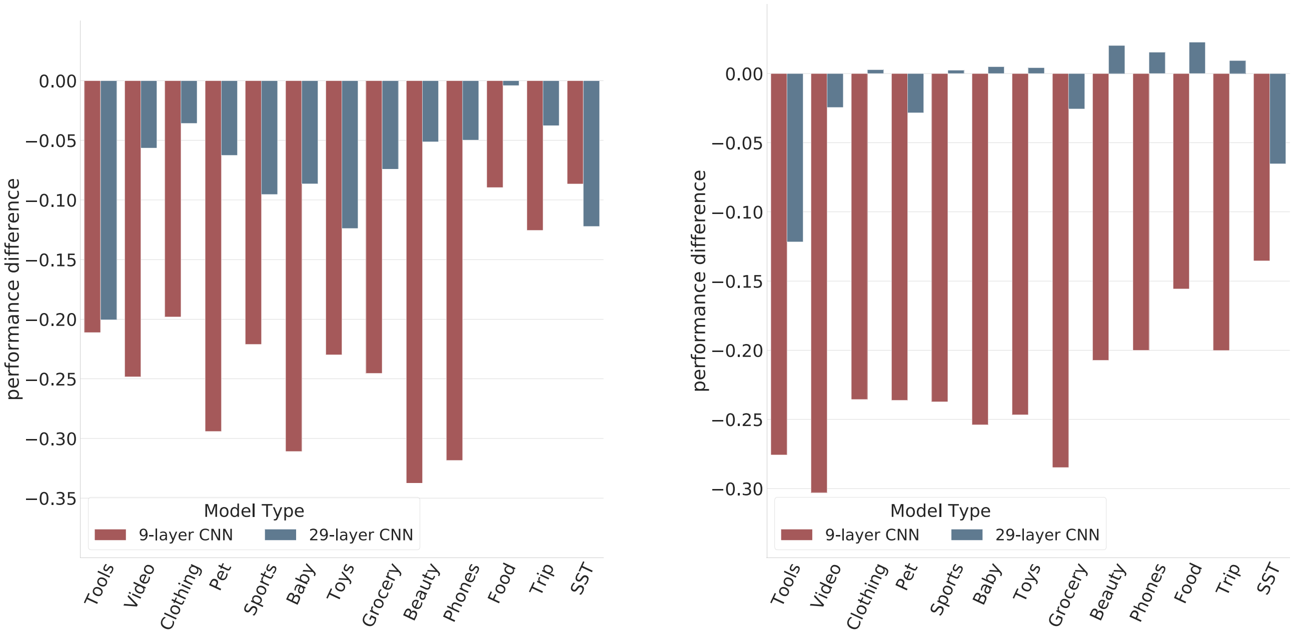
FIGURE 7. **Performance differences between word-and character-level inputs.**

deeper, the character-level input resulted in a better classification performance. All datasets reported higher AUROCs with character-level inputs than with word-level inputs for the 29-layer CNN. As repeated convolution and pooling processes can include a wider area in image recognition, a deeper CNN model can include a long sequence in text processing. As the character-level input is significantly higher than the word-level input, increasing the layers can aid the model in understanding the long-range relationship within the text, which consequently results in improved performance.

In contrast to the CNN-based models, the performance differences between the word- and character-level inputs for RNN-based models are consistent; the word-level inputs outperformed the character-level inputs. In particular, the performance difference between the two input levels are obvious for the vanilla RNN structure; the word-level input significantly outperformed the character-level input and the difference is at least greater than 0.1. As the character-level input has a significantly longer input sequence than the word-level input for the same sentence or text, most information provided in the beginning portion of the text is lost with the vanilla RNN, and the classification result is highly dependent on the ending portion of the text; consequently, the performance can become degraded as the sequence length increases. As LSTM and GRU have a mechanism to adaptively forget or remember the historical information inside the hidden state, important information in the beginning portion of the text can be delivered to the last hidden state to prevent performance degeneration. As the long-term memory module does not function for both LSTM and GRU, the word-level input yielded a higher performance than the character-level input; however, the difference between the two input levels are not as obvious as that with the vanilla RNN. When the bidirectional structure was adopted, the long-term dependency problem was alleviated considerably; however, the performance improvement from using the word-level input over the character-level input was not comparable to that with forward-directional LSTM and GRU.

3) PERFORMANCE COMPARISON BY MODEL

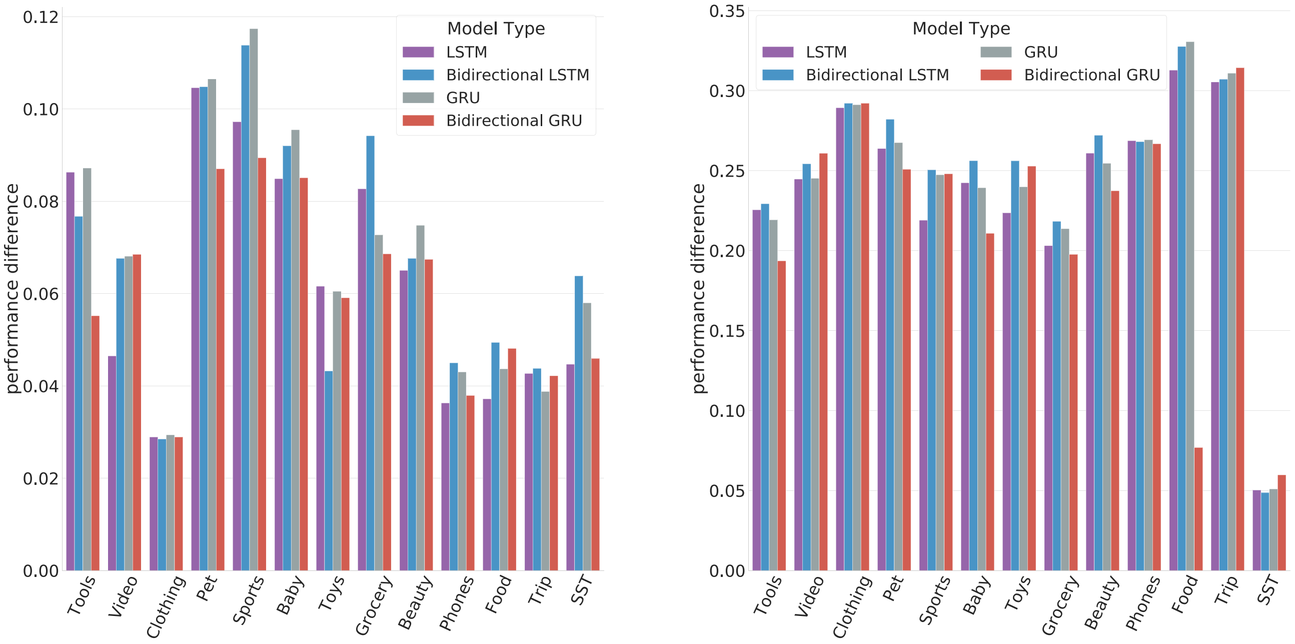
Figure 8 shows the performance improvements of the nine-layer CNN and 29-layer CNN over the one-layer CNN for the word-level input (a) and character-level input (b). When the word-level input was used, increasing the number of layers did not improve the sentiment classification performance, as shown in Figure 8a. Consequently, both the nine- and 29-layer CNNs resulted in lower AUROCs than the one-layer CNN. However, we could not conclude that the number of CNN layers and the sentiment classification performance exhibited a consistent trend. The nine-layer CNN yielded the worst performance among the three CNN models, followed by the 29-layer CNN and the one-layer CNN. When the character-level input was used, the nine-layer CNN showed a similar tendency for the word-level input; moreover, the sentiment classification performance deteriorated significantly. Conversely, the 29-layer CNN yielded a higher AUROC than the one-layer CNN for eight datasets among the 13 (61.5%). To summarize, if the CNN is adopted for text



(a) Using word-level input

(b) Using character-level input

FIGURE 8. **Performance differences between each CNN-based model and the one-layer CNN.**



(a) Using word-level input

(b) Using character-level input

FIGURE 9. **Performance differences between each RNN-based model and the vanilla RNN.**

classification because of its computational efficiency or ability for parallelism, either a shallow CNN with a word-level input or a deeper CNN with a character-level input should be considered.

Figure 9 shows the performance improvement of LSTM, bidirectional LSTM, GRU, and bidirectional GRU over the vanilla RNN for word-level inputs (a) and character-level inputs (b). Generally, increasing the structural complexity of the RNN-based models is effective for improving the sentiment classification performance. First, employing the LSTM or GRU module aids in improving the classification performance, especially for character-level inputs. As the AUROCs of the vanilla RNN with character-level inputs were significantly lower than those with word-level inputs for the same dataset, the magnitude of performance improvement was significant for character-level inputs. However, the absolute AUROCs of word-level inputs with LSTM and GRU were generally higher than those of character-level inputs with LSTM and GRU. In addition, we could not conclude that either LSTM or the GRU was more effective than the others in our experiment. Next, employing the bidirectional structure yielded additional performance enhancement in most datasets, although the improvements were marginal at times when compared with the improvement in LSTM/GRU over the vanilla RNN.

V. CONCLUSION

In this study, we conducted a comparative experiment on various deep-learning-based sentiment classification models. Based on previous studies, we selected three CNN structures and five RNN structures. They were compared in terms of AUROC using 13 datasets. Furthermore, we investigated the effects of two different input levels (word and character levels) on the classification performance.

Based on the experimental results, the conclusions are as follows. First, sentiment classification performances improved in accordance with the training dataset volume, irrespective of the model structure; the larger the dataset size, the better was the classification performance. Next, the input levels imposed different effects on the CNN-based and RNN-based models. Primarily, the word-level input yielded a better classification performance than the character-level input for the same model structure. However, we observed that the character-level input with a deep CNN structure occasionally yielded the better performance among the CNN structures with two input levels. Nonetheless, this was not applicable to the RNN-based models; the word-level input always outperformed the character-level input. Subsequently, increasing the model complexity yielded different effects on CNN-based and RNN-based models. For the CNN-based models, we could not conclude that increasing the model complexity would consistently improve or degenerate the classification performance. However, for the RNN-based models, increasing the model complexity always improved the performance. Employing an LSTM/GRU unit improved the performance; furthermore, the bidirectional structure provided an additional improvement to the forward-directional RNN. Among the individual model and input-level combinations, the bidirectional LSTM with word-level inputs yielded the best performance for five out of 13 datasets, followed by the GRU with word-level inputs (three datasets) and the one-layer CNN with word-level inputs (two datasets).

Although the experimental results provided certain practical implications for building a sentiment classification model, certain limitations existed in the current work, which motivated us to plan for future research directions. To compare the fundamental performance of deep-learning models, words and characters were randomly initialized and trained together with network parameters in our study. However, certain pretrained word vectors, such as bidirectional encoder representations from transformers, reported the best performance in many NLP tasks; furthermore, comparing the performances between word vectors trained from scratch and pretrained word vectors would be beneficial.

REFERENCES

[1] A. L. Fruhling and L. A. Digman, "The impact of electronic commerce on business-level strategies," *J. Electron. Commerce Res.*, vol. 1, no. 1, pp. 13–22, 2000.

[2] J. Kim and J. Park, "A consumer shopping channel extension model: Attitude shift toward the online store," *J. Fashion Marketing Manage.*, vol. 9, no. 1, pp. 106–121, Mar. 2005.

[3] A. Salinca, "Business reviews classification using sentiment analysis," in *Proc. 17th Int. Symp. Symbolic Numeric Algorithms Sci. Comput. (SYNASC)*, Sep. 2015, pp. 247–250.

[4] W. Duan, B. Gu, and A. B. Whinston, "Do online reviews matter?—An empirical investigation of panel data," *Decis. Support Syst.*, vol. 45, no. 4, pp. 1007–1016, 2008.

[5] Y. Chen and J. Xie, "Online consumer review: Word-of-mouth as a new element of marketing communication mix," *Manage. Sci.*, vol. 54, no. 3, pp. 477–491, Mar. 2008.

[6] C. Dellarocas, X. M. Zhang, and N. F. Awad, "Exploring the value of online product reviews in forecasting sales: The case of motion pictures," *J. Interact. Marketing*, vol. 21, no. 4, pp. 23–45, Jan. 2007.

[7] F. Zhu and X. M. Zhang, "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics," *J. Marketing*, vol. 74, no. 2, pp. 133–148, Mar. 2010.

[8] F. Tang, L. Fu, B. Yao, and W. Xu, "Aspect based fine-grained sentiment analysis for online reviews," *Inf. Sci.*, vol. 488, pp. 190–204, Jul. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025519301872

[9] M. López, A. Valdivia, E. Martínez-Cámara, M. V. Luzón, and F. Herrera, "E^{2}SAM: Evolutionary ensemble of sentiment analysis methods for domain adaptation," *Inf. Sci.*, vol. 480, pp. 273–286, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025518309873

[10] D. Hyun, C. Park, M.-C. Yang, I. Song, J.-T. Lee, and H. Yu, "Target-aware convolutional neural network for target-level sentiment analysis," *Inf. Sci.*, vol. 491, pp. 166–178, Jul. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025519302877

[11] H. H. Do, P. Prasad, A. Maag, and A. Alsadoon, "Deep learning for aspect-based sentiment analysis: A comparative review," *Expert Syst. Appl.*, vol. 118, pp. 272–299, Mar. 2019.

[12] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, 2008.

[13] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 4, 2018, Art. no. e1253.

[14] S. L. Sonawane and P. V. Kulkarni, "Extracting sentiments from reviews: A lexicon-based approach," in *Proc. 1st Int. Conf. Intell. Syst. Inf. Manage. (ICISIM)*, Oct. 2017, pp. 38–43.

[15] M. Lailiyah, S. Sumpeno, and I. K. E. Purnama, "Sentiment analysis of public complaints using lexical resources between Indonesian sentiment lexicon and Sentiwordnet," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, Aug. 2017, pp. 307–312.

[16] L. Gatti, M. Guerini, and M. Turchi, "SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis," *IEEE Trans. Affective Comput.*, vol. 7, no. 4, pp. 409–421, Oct. 2016.

[17] Suhariyanto, A. Firmanto, and R. Sarno, "Prediction of movie sentiment based on reviews and score on rotten tomatoes using SentiWordnet," in *Proc. Int. Seminar Appl. Technol. Inf. Commun.*, Sep. 2018, pp. 202–206. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8549704

[18] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Lang. Resour. Eval.*, vol. 39, nos. 2–3, pp. 165–210, May 2005.

[19] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proc. Int. Conf. Lang. Resour. Eval. (LREC)*, vol. 10, 2010, pp. 2200–2204.

[20] Y. Dang, Y. Zhang, and H. Chen, "A lexicon-enhanced method for sentiment classification: An experiment on online product reviews," *IEEE Intell. Syst.*, vol. 25, no. 4, pp. 46–53, Jul. 2010.

[21] M. R. Saleh, M. Martín-Valdivia, A. Montejo-Ráez, and L. Ureña-López, "Experiments with SVM to classify opinions in different domains," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14799–14804, Nov. 2011.

[22] Z. Zhang, Q. Ye, Z. Zhang, and Y. Li, "Sentiment classification of Internet restaurant reviews written in Cantonese," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7674–7682, Jun. 2011.

[23] M. Heikal, M. Torki, and N. El-Makky, "Sentiment analysis of Arabic Tweets using deep learning," *Procedia Comput. Sci.*, vol. 142, pp. 114–122, Oct. 2018.

[24] H. Ghulam, F. Zeng, W. Li, and Y. Xiao, "Deep learning-based sentiment analysis for roman urdu text," *Procedia Comput. Sci.*, vol. 147, pp. 131–135, Jan. 2019.

[25] A. S. M. Alharbi and E. De Doncker, "Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information," *Cognit. Syst. Res.*, vol. 54, pp. 50–61, May 2019.

[26] K. Chakraborty, S. Bhattacharyya, R. Bag, and A. E. Hassanien, "Comparative sentiment analysis on a set of movie reviews using deep learning approach," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.* Cairo, Egypt: Springer, 2018, pp. 311–318.

[27] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," 2017, *arXiv:1708.00524*. [Online]. Available: https://arxiv.org/abs/1708.00524

[28] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *CSURACM Comput. Surv.*, vol. 50, no. 5, pp. 1–22, Sep. 2017.

[29] B. Liu, "Sentiment analysis and subjectivity," in *Handbook of Natural Language Processing*, vol. 2, no. 2010. Boca Raton, FL, USA: CRC Press, 2010, pp. 627–666.

[30] Y. Wilks and J. Bien, "Beliefs, points of view, and multiple environments," *Cognit. Sci.*, vol. 7, no. 2, pp. 95–119, Apr. 1983.

[31] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proc. 2nd Int. Conf. Knowl. Capture*, 2003, pp. 70–77.

[32] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2003, pp. 129–136.

[33] N. U. Pannala, C. P. Nawarathna, J. Jayakody, L. Rupasinghe, and K. Krishnadeva, "Supervised learning based approach to aspect based sentiment analysis," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Dec. 2016, pp. 662–666.

[34] R. K. Palkar, K. D. Gala, M. M. Shah, and J. N. Shah, "Comparative evaluation of supervised learning algorithms for sentiment analysis of movie reviews," *Int. J. Comput. Appl.*, vol. 142, no. 1, pp. 20–26, 2016.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[39] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, "Convolutional recurrent neural networks for text classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6.

[40] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," 2016, *arXiv:1603.03827*. [Online]. Available: https://arxiv.org/abs/1603.03827

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learn*. Cambridge, MA, USA: MIT Press, 2016.

[42] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.

[43] 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, pp. 2359–2364.

[44] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*. [Online]. Available: https://arxiv.org/abs/1404.2188

[45] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: https://arxiv.org/abs/1408.5882

[46] 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.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[48] H. T. Le, C. Cerisara, and A. Denis, "Do convolutional networks need to be deep for text classification?" in *Proc. Workshops 22nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[49] 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.

[50] C. Wang, M. Zhang, S. Ma, and L. Ru, "Automatic online news issue construction in Web environment," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 457–466.

[51] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, and C. Bizer, "DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[52] J. Mcauley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst. (RecSys)*, 2013, pp. 165–172.

[53] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," 2016, *arXiv:1606.01781*. [Online]. Available: https://arxiv.org/abs/1606.01781

[54] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[56] K. Cho, B. Van Merriënboer, 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: https://arxiv.org/abs/1406.1078

[57] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: https://arxiv.org/abs/1412.3555

[58] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[59] J. Xu, D. Chen, X. Qiu, and X. Huang, "Cached long short-term memory neural networks for document-level sentiment classification," 2016, *arXiv:1610.04989*. [Online]. Available: https://arxiv.org/abs/1610.04989

[60] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," *Neurocomputing*, vol. 308, pp. 49–57, Sep. 2018.

[61] Y. Yin, Y. Song, and M. Zhang, "Document-level multi-aspect sentiment classification as machine comprehension," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2044–2054.

[62] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1422–1432.

[63] R. Hu, L. Rui, P. Zeng, L. Chen, and X. Fan, "Text sentiment analysis: A review," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2018, pp. 2283–2288.

[64] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, *arXiv:1702.01923*. [Online]. Available: https://arxiv.org/abs/1702.01923

[65] P. Singhal and P. Bhattacharyya, "Deep learning for sentiment analysis: A survey," Center Indian Lang. Technol., Indian Inst. Technol., Mumbai, India, Tech. Rep., 2016.

[66] T. Katić and N. Milićević, "Comparing sentiment analysis and document representation methods of Amazon reviews," in *Proc. IEEE 16th Int. Symp. Intell. Syst. Inform. (SISY)*, Sep. 2018, pp. 283–286.

[67] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[68] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," 2016, *arXiv:1605.05101*. [Online]. Available: https://arxiv.org/abs/1605.05101

[69] Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*. [Online]. Available: https://arxiv.org/abs/1801.02143

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1412.6980

[71] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

SEUNGWAN SEO received the B.S. degree in information and technology management from SeoulTech. He is currently pursuing the Ph.D. degree in industrial and management engineering with Korea University, Seoul, South Korea. His research interest is in developing machine-learning (especially deep learning) algorithms for unstructured data, such as images and text.

CZANGYEOB KIM received the B.S. degree in computer science from Kookmin University and the M.S. degree in information security from Sungkyunkwan University. He is currently pursuing the Ph.D. degree with the School of Industrial Management Engineering, Korea University, South Korea. His main research interest is in developing machine-learning algorithms and applying them to solve engineering problems in cybersecurity, computer networks, and healthcare.

HAEDONG KIM received the B.S. and M.S. degrees in industrial and management engineering from Korea University, Seoul, South Korea. He is currently pursuing the Ph.D. degree in industrial and manufacturing engineering with The Pennsylvania State University, State College, PA, USA. His research interest is data analytics based on sensor networks and optimization with special focus on healthcare analytics and smart additive manufacturing systems.

KYOUNGHYUN MO received the B.S. degree in industrial engineering from the Seoul National University of Science and Technology, Seoul, South Korea, and the M.S. degree in industrial and management engineering from Korea University. He is currently with SK C&C. His current research interests include deep learning and natural language processing.

PILSUNG KANG received the B.S. and Ph.D. degrees in industrial engineering from Seoul National University. He is currently an Associate Professor with the School of Industrial Management Engineering, Korea University, South Korea. His main research interest is in developing machine-learning algorithms for both structured and unstructured data (image, video, and text) and applying them to solve engineering and business problems, such as fault classification in manufacturing, abnormal behavior detection from system logs, and sentiment classification from news and review texts.