IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 26, 2020, accepted April 28, 2020, date of publication May 4, 2020, date of current version May 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992013

Deep Learning-Based Sentiment Classification: A Comparative Survey

ALHASSAN MABROUK[1], REBECA P. DÍAZ REDONDO[2], AND MOHAMMED KAYED[3]
[1]Mathematics and Computer Science Department, Faculty of Science, Beni-Suef University, Beni Suef 62511, Egypt
[2]Information & Computing Lab, AtlanTTIC Research Center, Telecommunication Engineering School, Universidade de Vigo, 36310 Vigo, Spain
[3]Computer Science Department, Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni Suef 62511, Egypt

Corresponding author: Mohammed Kayed (mskayed@gmail.com)

This work was supported in part by the European Regional Development Fund (ERDF) and the Galician Regional Government through the agreement for funding the Atlantic Research Center for Information and Communication Technologies (AtlantTIC), and in part by the Spanish Ministry of Economy and Competitiveness through the National Science Program under Grant TEC2017-84197-C4-2-R.

ABSTRACT Recently, Deep Learning (DL) approaches have been applied to solve the Sentiment Classification (SC) problem, which is a core task in reviews mining or Sentiment Analysis (SA). The performances of these approaches are affected by different factors. This paper addresses these factors and classifies them into three categories: data preparation based factors, feature representation based factors and the classification techniques based factors. The paper is a comprehensive literature-based survey that compares the performance of more than 100 DL-based SC approaches by using 21 public datasets of reviews given by customers within three specific application domains (products, movies and restaurants). These 21 datasets have different characteristics (balanced/imbalanced, size, etc.) to give a global vision for our study. The comparison explains how the proposed factors quantitatively affect the performance of the studied DL-based SC approaches.

INDEX TERMS Review mining, sentiment classification, neural networks, deep learning.

I. INTRODUCTION

Websites usually provide services for different purposes such as booking services [1], social media communication [2], political, E-Commerce (EC) and blogging. These sites allow users to share their viewpoints/opinions in forums [3], blogs [3], [4], reviews [1], news articles [3], [5], [6] or wikis. Consequently, they are considered as one of the most important sources of consumers (users) opinions. Within this context, a new research area, Sentiment Analysis (SA), has emerged with the aim of analyzing and categorizing user opinions. Consequently, SA is a relevant source of information for different application fields, such as sales measurement, political movements or prediction of election results. In fact, both individuals and organizations are interesting in SA. For instance, in the EC field, customers of big companies such as Amazon, Walmart or eBay (among others) seek opinions from other users before making a purchase. Additionally, companies carefully check and process these opinions (complaints or positive reviews) to enhance their products and to take strategic decisions for stocks and, even, production. However, these opinions are written in natural language, using colloquial forms, jargon and include abbreviations, lack of capitals, encounter problems with spelling, punctuation and grammar errors. Moreover, texts usually include other elements such as URLs, tags and other unstructured data. This makes it easier for users to read, but more difficult for machines to process; so, it is a challenge that requires a combination of different techniques to be managed. In fact, customer reviews must be pre-processed in order to be converted into structure data before being analyzed.

In review mining, many approaches have been proposed [7] with different SA tasks: aspect extraction [8], [9], Sentiment Classification (SC), ambiguous text [6], subjectivity classification [10] or opinion spam [11]. These approaches have worked on different domains [12], [13] or languages [3], [14], [15]. Among all of these approaches, SC is considered as one of the key tasks. In SC, the main goal is inferring the polarity of a given message or review [4], [16], [17]. That is, the text, document, sentence or feature is analyzed in order to know if it is positive, negative or neutral. Some websites allow users to directly

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

evaluate items or services using stars [3], numbers or a thumb up/down [18], [19]. However, customers also have the opportunity to write his/her opinion to supplement the assessment. Therefore, even in this case, it is still necessary to extract knowledge and polarity from customer opinions in order to identify ambiguous text, detect opinion spam or infer knowledge for strategic decisions.

In the specialized literatures, there are different proposals for Sentiment Classification (SC) of textual reviews. These approaches are usually classified into three types [20]: Lexicon [21], Machine Learning (ML) [4] and Hybrid-based approaches [22]. Lexicon-based are fast in training, but ML-based ones achieve state-of-the-art performance in SC. Hybrid-based proposals are characterized by its high complexity, so they are not popular yet. Consequently, ML-based approaches are, without doubt, the most popular for SC with models that have been repeatedly used in the literatures such as Naïve Bayes, Maximum entropy, Decision trees, Support Vector Machines or Neural Networks (NN). In fact, the last one, NN, is widely used [23]–[26] because of its higher efficiency (high performance and fast execution) as compared to the other alternatives.

Within the NN field, Deep Learning (DL) approaches have made a breakthrough in SA [25], so different researchers have analyzed DL-based SA proposals. In [26], the authors compare the performance of different DL methods for different SA tasks. However, in [27], the authors focus only on one of them: a deep comparison of the performance of aspect extraction.

To the best of our knowledge, there is no any prior comparative study of performance about the application of DL in Sentiment Classification, although many approaches have been proposed on different levels of SC: aspect [28], [29]; sentence [30], [31]; and document [32], [33]). Therefore, this paper provides a comparative and a comprehensive literature-based survey for the existing DL approaches that are suggested to solve the task of SC. More specifically, our study targets proposals within the field of DL-based SC for customer reviews.

Consequently, the paper has two main contributions. First, it discusses the utmost key factors that affect the performance of DL-based SC approaches. These factors are classified into three different categories: data preparation based factors, feature representation based factors and the techniques based factors. Second, it compares the performance of more than 100 DL-based approaches on SC that have been published in the specialized literatures. These approaches have been assessed on 21 widespread datasets of three specific review domains (products, movies and restaurants). All datasets have different characteristics (balanced/imbalanced, size, etc.), which gives a global vision for our study.

The remainder of the paper is organized as follows. Section II shortly presents a background knowledge about neural network and DL. Section III discusses the factors that affect the performance of DL-based SC approaches. The set-up of the experiments are reported in Section IV. The results of recent DL-based SC approaches on different domains are presented and compared in Section V. Extra factors that could be used to enhance the performance of DL-based SC approaches as well as open issues on SC are presented in Section VI. Finally, Section VII concludes our work and expounds our proposal for the future work.

II. NEURAL NETWORK

Neural Networks (NNs) have different advantages: (i) NNs provide a nonlinear model, which is flexible in representing complex relationships; (ii) NNs are able to estimate the posterior probabilities (i.e., performing statistical analysis); (iii) the execution time of an NN model is not excessively long; and (iv) NNs give good performance even with noisy data. Therefore, NNs have been widely used in sentiment analysis. In this section, we provide a brief introduction about NNs and DL, and how to train DL models.

A. NN ARCHITECTURE

An NN is a network diagram that interconnects nodes, also known as neurons, which transmit data among them and are arranged in layers, as shown in Figure 1. Each neuron in the network has a certain random weight. Also, each layer has a bias that influences the output. A neuron is a simple mathematical model that calculates an output value in two steps [4]. First, it calculates an input function as the sum of weights and the values of the input neurons. Second, it applies an activation function to this sum to provide the output. The activation function (e.g., Sigmoid, Tanh or Relu) is typically a nonlinear function. Relu activation function is the easiest to compute, the fastest to converge in training and yields equal or better performance in NNs [34].

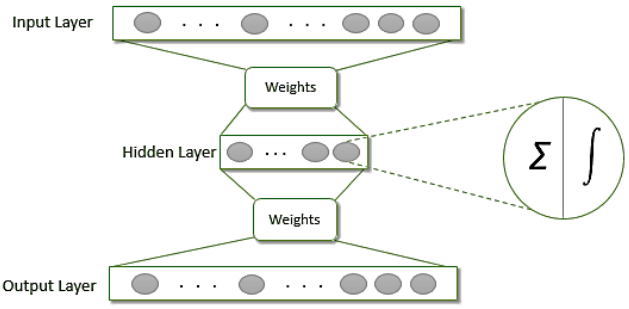

FIGURE 1. Neural network architecture.

The architecture of NN includes three layers: input, hidden and output layer. The input layer is a *word embedding vector,* which will be discussed later. Secondly, the hidden layer takes the input features and gives an output to the next layer using the previously mentioned activation function. Finally, for each class in the output layer, there is a probability distribution via a softmax function.

B. DEEP LEARNING: A BRIEF OVERVIEW

Despite the previously mentioned advantages of NNs, there is a clear shortcoming for SC applications. When the number

of input and hidden nodes increases, an overfitting problem occurs due to the increase of parameters within the NN. To address this problem, deep learning (DL) methods reduce the number of parameters, but they include multiple layers. The connection weights among neurons can be adjusted to perform tasks such as classification with the aim of achieving high performance in SC [35], [36].

DL models are trained, i.e. the weights among neurons are adjusted, by using a backpropagation process [37] in which a loss function is minimized using the Stochastic Gradient Descent (SGD) algorithm. Gradients of the loss function calculate weights from the last hidden layer to the output layer. Then, these weights are recursively calculated by applying the chain rule backwards. SGD is an iterative refinement process that is stopped when certain stopping criteria are met. It estimates the parameters for each training sample as opposed to the whole set of training samples in batch gradient descent. For example, the training values are loaded into the input nodes. If misclassification occurs, the error is propagated back through the network, modifying the weights to minimize the error.

It is necessary to remark that DL models have a high computational cost, such as high memory usage, for instance. However, DL models provide very good results at SA tasks [38], [39]. In fact, different DL models have been used to solve the SC problem, including Convolutional [40], Recurrent [32], [33], [41], Recursive [42], [43] and Hybrid Neural Networks [25], [44].

III. PERFORMANCE OF DL-BASED SC APPROACHES

This section addresses the factors that affect the performance of DL-based SC approaches. As shown in Figure 2, these factors are classified into three different categories: data preparation based factors, feature representation based factors and classification techniques based factors. The next three subsections will discuss these three categories in details.

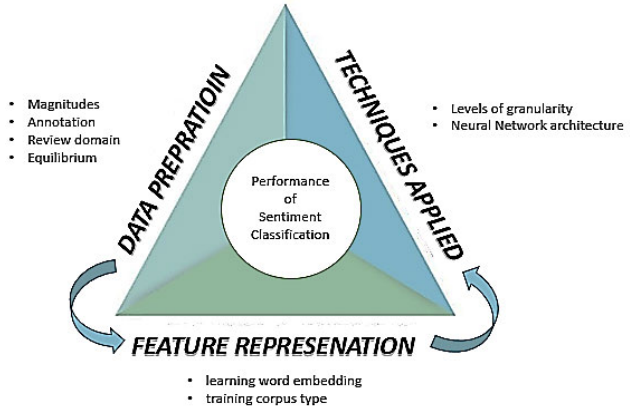

FIGURE 2. Factor that affect the performance of DL-based SC approaches.

A. DATA PREPARATION

Data or Corpora preparation is one of the fundamental challenges to generate useful information. In our case, the process of data preparation always starts by extracting users' reviews from the websites, handling the gaps in these reviews and converting them into a useable format such as the one in Figure 3.

Corpora are usually classified into three different types: polarity, subjectivity and ironic [6], [12]. A polarity corpus classifies the reviews into a specific number of categories (positive, negative, etc.). A subjective corpus extracts opinions from reviews or texts; for instance, Pang and Lee [10] used a cut-based classification to determine the subjectivity. Finally, an ironic corpus, the most complex in this field, extracts irony from the reviews [5], [6], [45]. Our contribution only focuses on the first one, Polarity or Sentiment Corpora, but expressed at different levels: documents [16], [46], [47], [4], sentences [3], [17] or aspects [48], [49]. The last one is commonly used in SC.

We suggest that four factors could affect the performance of SC during the polarity corpus preparation (see Figure 2):

1) MAGNITUDES

the number of classes may be binary, ternary or multiple (scale) classes. The first one, *Binary corpora* (''positive'' and ''negative''), ignores the neutral class. Some approaches, like [4], [17], [22], [23], [50], directly consider the vagueness (neutrality) as noise. Others, like in [16] consider them as positive opinions, since there is no negativity in the review. The second one, *Ternary corpora,* splits the reviews into three classes: ''positive'', ''negative'' and ''neutral'' [51]. Finally, *Scale corpora*, matches the reviews with a number, following a scale similar to the star ratings in websites, like in [52].

2) ANNOTATION

Data is modeled in two groups: labeled data and unlabeled data. The labeled data is divided into known classes (e.g., positive and negative), unlike in the unlabeled data in which data labels are unknown. For instance, in [50], labeled data is split into three classes (negative, positive and neutral).

3) REVIEW DOMAIN

In our study, we focus on three application fields or review domains: products, movies and restaurants. In all of them, we analyze the behavior at the three considered levels: document-level [4], [10], [16]–[18], [14], [23], [52], [53], sentence-level [49] and aspect-level [5], [54]; which is explained in Figure 3.

4) EQUILIBRIUM

Corpora may be classified as balanced or imbalanced. On one hand, in balanced corpora, data are partitioned and distributed equally on different classes. For example, in [10], [17], the corpus includes 2,000 movie reviews from *rottentomatoes*, which are divided into two balanced categories (+ve and –ve). In [55], [56], the analysis is done using a multi-domain corpus of 400 reviews from *Epinion* that are distributed on 8 balanced categories of 50 reviews each (music, computers, movies, phones, books, cookware, cars and hotels). On the other hand, imbalanced corpora have

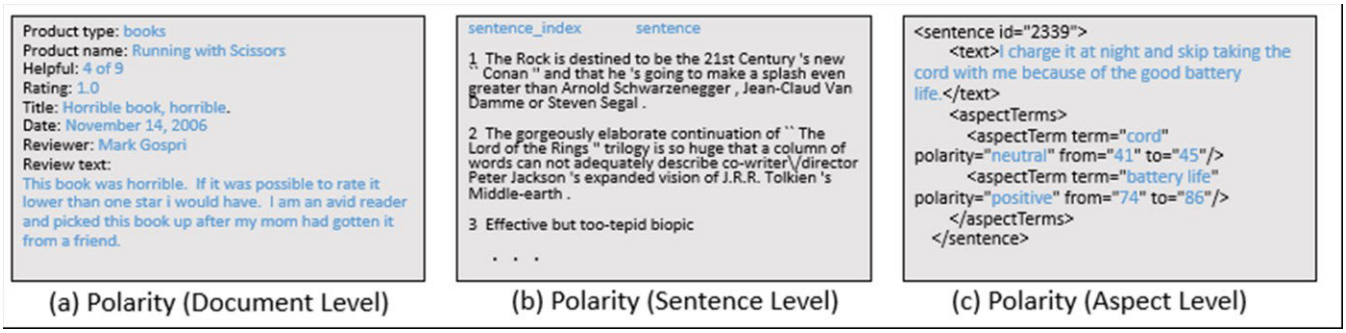

FIGURE 3. Tuples output of polarity structure reviews on different levels.

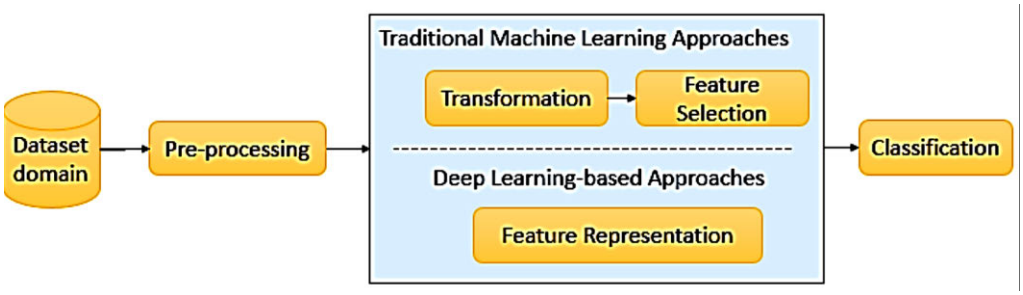

FIGURE 4. Comparison between traditional ML and recent DL-Based approaches for SC.

classes with significantly different number of elements. For example, SINAI dataset in [16] includes about 2,000 documents (reviews) of 10 camera models, where 93% were classified as positive. In Dave *et al.* [52], a corpus from CNET and Amazon with 7 imbalanced categories is used. However, Rushdi Saleh *et al.* [16] achieved promising results working with an imbalanced data set of product reviews.

B. FEATURE REPRESENTATION

DL models are oriented to automatically learn the best representations (features) to return a probability that could be used to assign a given label to each input [57], [58]. Figure 4 compares the architecture of (i) traditional ML-based approaches and (ii) DL-based approaches on SC. It is necessary to remark that in both cases. There is a previous stage, pre-processing, which is required to clean and organize the data. After that, the first approach, ML-based models, consists of three consecutive stages: transformation (e.g. Term Frequency—Inverse Document Frequency), feature selection (e.g., chi-square) and construction of the classifier. Some studies have demonstrated that working on the feature selection stage can potentially improve the classification accuracy [19]. After the pre-processing phase, the second approach, DL-based models, only uses feature representation (word embedding). This entails a more flexible and adaptable solution to the data [59]. Generally speaking, DL-based approaches give better performances than traditional ML-based approaches with large datasets, since DL-based models require large sets of training data.

Word embedding, typically applied for feature representation in DL-based approaches, transforms words or phrases into vectors of real numbers, which is also known as word vector representation or distributed representation. In short, word embedding applies a dimensionality reduction that gets a lower-dimensional dense vector space from a high-dimensional sparse embedding. Each dimension of the embedding vector represents a latent feature of a word. This means, it describes syntactic and semantic properties of the words [4], [60], [61].

This strategy was firstly proposed in [37] and it has been widely used in SA [62], [63]. Word embedding handles many SA tasks such as aspect term extraction [64], SC [32], [59], [65], [66], among other. The specialized literatures offer different alternatives (algorithms) to generate word embedding. For example, the Sentiment-Specific Word Embedding (SSWE) method is proposed in [67], and its features offer competitive results in context-based SA [60], but it gave a poor performance in Aspect-Based Sentiment Analysis (ABSA) [39]. In [32], the quality of representations is improved for longer documents. Additionally, a Bilingual Sentiment Word Embedding (BSWE) model for cross-language SC is proposed in [68].

We consider that the performance of SC during the feature representation phase can be affected by two factors, as depicted in Figure 2:

1) LEARNING WORD EMBEDDING

This is one of the most important factors that affects the performance of DL-based approaches. There are usually two

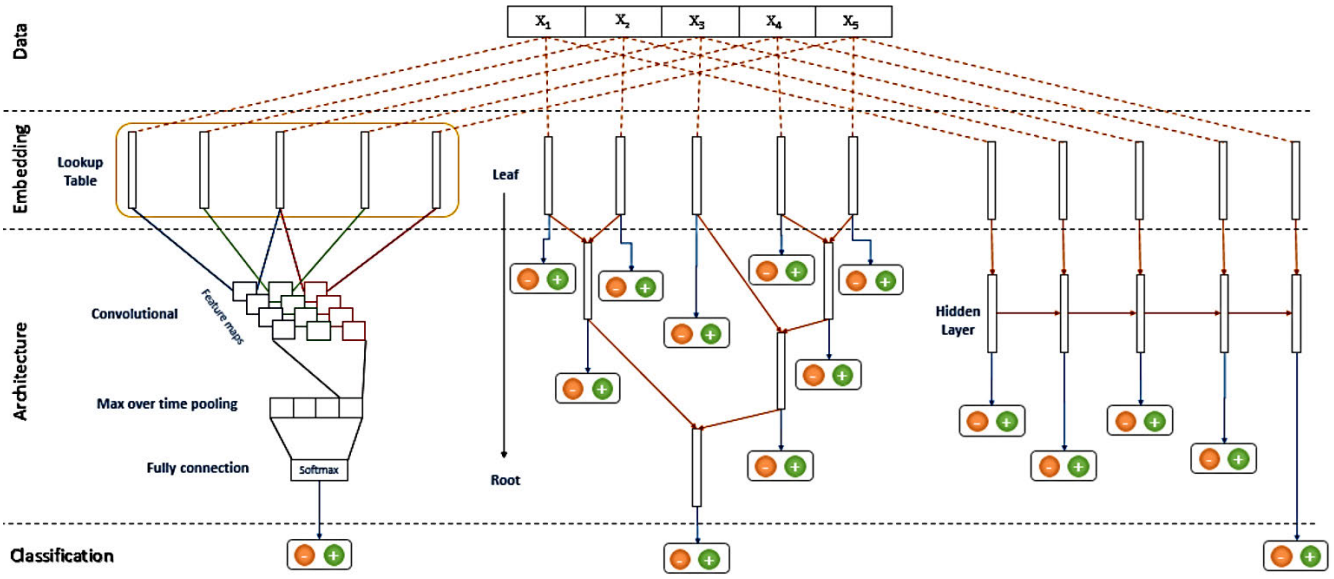

FIGURE 5. Convolutional neural network (left), recursive neural network (center), and recurrent neural network (right). The architectures for SC.

methods to create word embedding versions: matrix factorization and neural network.

The first approach, Matrix factorization-based word embedding is a linear algebra process that applies rank reduction on a large term-frequency matrix. This matrix measures co-occurrence of terms in two frequencies: the rows of Term-Document frequencies matrix represent words and the columns represent the documents or the paragraphs. On the other side, both rows and columns of the Term-Term frequencies matrix represent words. In [69], the authors propose applying word embedding with matrix factorization for personalized review-based rating prediction. Besides, there are many techniques that are based on observed co-occurrence patterns to reduce the dimensionality such as clustering, Latent Dirichlet Allocation (LDA) and Singular Value Decomposition (SVD) [70].

The second approach, Neural network-based word embedding has words as input and gives context as an output. For example, a Neural Probabilistic Language Model (NPLM) [57] is the first word embedding model with a shared lookup table. Given a word and its previous words, the model predicts the probability function for its next word. Wang and Xia [71] developed a neural architecture to train a sentiment bearing word embedding by integrating the sentiment supervision at both the document and the word levels.

2) TRAINING CORPUS TYPE

The performance of SC is clearly dependent on the corpora used. For instance, in [64], [63], some options are analyzed and the conclusion is that Amazon corpus has a better performance on SC than both Wikipedia and Google news corpora, because Amazon corpus contains more opinion words.

C. APPLIED CLASSIFICATION TECHNIQUES

As shown in Figure 2, taking into account the techniques applied for SC, there are two factors that could affect the performance: on one hand, the level of granularity and, on the other hand, the neural network architecture used. NNs have emerged as a good approach for SC, so we have focused only on this approach.

1) LEVEL OF GRANULARITY

Several DL-based approaches have been proposed to solve the SC problem at the three levels: document, sentence and aspect. Table 1 lists the most recent and common approaches on each one.

2) NEURAL NETWORK ARCHITECTURE

As depicted in Figure 5, a NN loads the words (X) into the input nodes (like word embedding) and the final values of output nodes give the classification results. Generally, it is accepted that there are four types of NN architectures that try to capture semantic and syntactic information: (i) Recursive Neural Networks (RecNN); (ii) Recurrent Neural Networks (RNN); (iii) Convolutional Neural Networks (CNN); and (iv) hybrid approaches. Table 1 classifies different approaches that can be found in the literatures according to the type of NN. Table 2 gives a brief comparison among these approaches summarizing the goal, the advantages and disadvantages of each one. Note that, hybrid approaches are not included in Table 2 because there are no common characteristics, since they depend on the types of NN that are combined.

In parallel, these models can be split into two types: feed-forward neural network (e.g., CNN) and backward neural network (e.g., RNN and RecNN). The four NN architectures

TABLE 1. Allocations of literatures based on SC-levels with the four types of NNs.

NN Architecture	Document	Sentence	Aspect
Convolutional	[23], [25], [75], [76]	[38], [40], [45], [62], [77], [78], [79], [80], [81], [82]	[83]
Recurrent	[25], [33], [84], [85], [86], [87], [88], [89], [90], [91]	[31], [65], [79], [80], [81], [92], [93], [94], [95], [96], [97], [98]	[99], [28], [29], [35], [39], [100], [101], [102], [103], [104], [105], [106], [107]
Recursive	--	[30], [36], [42], [43], [52], [108], [109], [110], [111], [112], [113]	[49]
Hybrid	[5], [25], [55], [114]	[44], [115], [116], [117]	--

(CNN, RNN, RecNN and Hybrid neural networks) will be discussed in the next four subsections, including information about the different DL methods.

a: CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN has become a popular DL model which is used by many researchers in various domains such as image processing and Natural Language Processing (NLP). In NLP, the words are converted into vectors [40], [59], [72], [75], [79]. CNN is characterized by the non-linearity of the model and its ability to learn embedding from small size regions. Its architecture is shown in Figure 5 (left) and it consists of four layers: an embedding (input) layer, a convolutional layer, a pooling layer and a fully connected (output) layer. In the first one, the embedding layer, each review is embedded at word-level and it is represented as a matrix. In the second one, the convolutional layer, the width of a filter is fixed to the dimensionality of the word vector in order to capture the relationships among adjacent words, producing a feature map. Thus, this layer (convolutional one) captures important features (n-gram) for each feature map (i.e., a specification of a semantic/structural feature) by directly applying word vectors. Both word vectors and the shared (word independent) kernels are the parameters of CNN, which can capture the predictive structures of small regions. In the third layer, pooling layer, the max-over-time pooling operation extracts the maximum value corresponding to each feature map. The pooling scheme can also be used to deal with the variant lengths of the feature maps produced by filters of different sizes. Besides, the max-pooling layer may produce a fixed-length output regardless of the size of the filter window. Last, in the fourth layer (output), the features are extracted and concatenated in the fully connected layer, which has a probability distribution over the output classes. It is worthy to notice that a deep CNN may have more than four layers: for instance, one input layer, two convolution layers, two max-pool layers and a fully connected layer with a softmax output. Experiments show that deep CNN models, even without any feature engineering or linguistic patterns, still outperform state-of-the-art models. Dos Santos and Gatti [75] focused on deep CNN for sentiment detection in short texts. Zhang *et al.* [24] showed that, for text classification, a deep CNN over characters performs properly.

The CNN model is applied in many tasks. Tang *et al.* [25] used hierarchical structure in SC. Guan *et al.* [79] proposed a novel DL framework for review SC that employs prevalently available ratings as weak supervision signals. Yu and Jiang [59] learned sentence embedding using two auxiliary tasks (whether the sentence contains a positive or a negative domain-independent sentiment word), which did not predict of pivot features on a large set. Zhang *et al.* [24] applied a character-level CNN for text classification and achieved competitive results. CNN can extract local n-gram features within texts but may fail to capture long-distance dependency, Long Short Term Memory (LSTM) can address this problem by sequentially modeling texts [76]. Many papers used multiple algorithms providing other advantages. CNN and RNN are often combined with sequence-based or tree-structured models [25], [40], [38], [41]. Wang *et al.* [76] used a regional CNN-LSTM to predict the valence arousal ratings of texts. LSTM was used in combination with CNN to perform SA for short text, which achieved better performance in terms of accuracy in [77]. The experiment showed that CNN is an alternative to overcome the high computational cost of NN, but it requires more training time compared with other techniques.

Below, we summarize some CNN models that have been applied on SC: DCNN [38], CNN [40], MVCNN [115], Seq2-bown-CNN [72], UPNN [116], CNN-Rule-q [117], Char-level CNN [24] and PF-CNN [80]:

DCNN: A Dynamic CNN that applies the dynamic k-max pooling strategy (i.e., selecting the k most active features) for sentence modeling.

CNN: A model that uses two channels for sentence classification. First, a static channel that is kept static throughout the training. Second, a non-static channel that is fine-tuned for each task. The multichannel one applied the two channels for each filter.

MVCN: A multichannel variable-size CNN that introduces multichannel embedding for sentence classification.

Seq2-bown-CNN: It groups texts information as a sequence. It has two sequential convolution layers and another convolutional layer to represent the entire document with a bag of words.

UPNN: It learns the semantic representations of user and product in a bottom-up way.

TABLE 2. Advantages and disadvantages of the most common NN models on SC.

NN Architecture	Goal	Advantages	Disadvantages
Convolutional	The essence in CNN are (i) the non-linearity of the model and (ii) the ability to learn embedding for small fixed size regions.	It overcomes the high computational cost of NNs.	It requires more training time compared with other techniques
Recurrent	RNN can capture sequential information in flexible computational steps.	It reduces the number of parameters needed to learn.	The output of one state depends on the previous state. Thus, it needs from huge amount of memory.
Recursive	RecNN is a generalization of RNN that applies recursively the same set of weights over a directed acyclic, but in a tree structure input.	It elegantly learns compositional semantic in simpler structure.	The application on SC still requires further research, which leads to inaccuracies.

PF-CNN: The authors use the information of aspects on CNN by parameterized filters and parameterized gates.

3W-CNN: This method helps to reclassify the predictions by three ways to improve CNN by Naive Bayes SVM.

CNN-Rule-q: This method changes the structural information of the logical rules to weight the NN through an iterative distillation method.

b: RECURRENT NEURAL NETWORK (RNN)

RNN has become popular in SC because it can capture sequential information in flexible computational steps. The RNN model has two important features compared to the CNN. First, CNN has different parameters at each layer, while the parameters in RNN are the same at each step (i.e., it reduces the number of parameters needed to learn). Second, in RNN, the output of one stage depends on the previous stage, thus it needs a huge memory. Therefore, RNN is more superior in processing sequential information compared to CNN.

Therefore, RNN is a robust network architecture (Figure 5, right) for processing sequential data. It allows cyclical connections and reuses the weights across different instances of neurons, each of them associated with different time steps. This idea can explicitly support the network to learn the entire history (i.e., current states) of the previous states. With this property, RNN maps an arbitrary length sequence to a fixed length vector.

The simple RNN has limitations caused by the gradient, making it difficult during the training in the backpropagation process. The two main problems are: (i) the gradient vanishing problem (i.e., the gradient comes close to zero) and (ii) the exploding gradient problem (i.e., being extremely high) which makes the learning process unstable. Tuning the parameters may improve the gradient. This limitation was reduced by the introduction of networks such as Long Short-Term Memory (LSTM) [118] and Gated Recurrent Units (GRU) [119]:

LSTM takes the whole document as a single sequence and the average of the hidden states of all words is used as a feature for classification. LSTM cannot extract the aspect information, this is why it achieves less performance. Therefore, many upgrades showed in [28], [100], [29], [39], [83], [120]. LSTM generally outperforms that of the CNN model [73], [84]. Wang *et al.* [62] proposed encoding entire tweets with LSTM, whose hidden state is used for predicting sentiment polarity. Qian *et al.* [92] proposed linguistically regularized LSTMs for SA with sentiment lexicons, negation words and intensity words.

GRU is similar to LSTM and it has not a memory unit. Cheng *et al.* [121] proposed a bidirectional GRU model to attend the aspect information for one given aspect and extract sentiment for that given aspect.

Additionally, there are extensions of RNN such as Bidirectional Recurrent Neural Networks (BRNN) [122]. BRNN incorporates a forward and a backward layer in order to learn information from preceding and following tokens. Also, Gate Recurrent Neural Networks (GRNN) [25] handles the document level SC, which obtain hierarchical representations by firstly building representations of sentences, and then aggregating those into a document representation. Moreover, in [90], BRNN is combined with LSTM to result in Bidirectional LSTM (BLSTM), which can access the long-range context in all input directions and more structure information. Ruder *et al.* [100] created a hierarchical BLSTM with a sentence level which is used as the input of a review level. Thus, BLSTM allows them to take into account inter- and intra-sentence context. They used only word embedding to make their system less dependent on extensive feature engineering or manual feature creation.

However, RNN has a remarkable problem processing the information (in the traditional encoder-decoder framework) that may entail the encoding of irrelevant information. One possible solution is to employ an *attention mechanism*, which allows the model to learn on which part of the text must focus. As an example, the work in [29] cannot know important

words, because it did not use this attention mechanism. Tang *et al.* [39] kept attention on aspect phrases. The general idea of the attention mechanism is to compute an attention weight from each lower level and, then, aggregate the weighted vectors for the higher level representation. This mechanism is suitable to be applied to SC, since it can focus on the important parts of the sentence. In the attention mechanism, the local semantic attention represents the implementation [84] by introducing a hierarchical network with two levels of attention mechanisms for document classification: word attention and sentence attention. Attention-based methods in [85] use attention mechanisms to build representations by scoring input words or sentences differently. Consequently, it is able to learn distributed document representations in Chinese and English. Also, this issue is solved by using external memory such as the *Memory Networks model* (MemNet). Memory networks have played a major role in ABSA task [35], [103], [96]. Recent researches show that memory network obtains the state-of-the-art results in SC [86]. Recurrent attention models have achieved superior performance [104] by learning a deep attention over the single level attention. Thus, multiple passes (or hops) over the input sequence could refine the attended words again and again to find the most important ones. In fact, Tang *et al.* [60] adopted a memory network (MemNet) solution based on multiple-hop attention.

RNN models have been widely used in SC, such as those using LSTM (e.g., Cached LSTM [83], TC-LSTM [39], TD-LSTM [29], ATAE-LSTM [28], HP-LSTM [100], and AF-LSTM [120]), those using attention mechanism (e.g., HN-ATT [84] and Structured Att [123]) and those using memory network (e.g., MemNet [35], DMN [99]). Skip-thought [89] and Byte mLSTM [32] used RNN for a word prediction. Other models are NCSL [31], Virtual Adversarial (VIRTUAL ADV) [65] and TopicRNN [33]:

Cached LSTM: It improves the LSTM ability to carry information for long distances.

NCSL: Neural Context-Sensitive Lexicon utilizes sentiment lexicons to treat the sentiment score of a sentence as a weighted sum of prior sentiment scores of negation words and sentiment words.

TD-LSTM: Target-dependent LSTM treats an aspect as a target by using two LSTM surrounding the aspect term.

TC-LSTM: Target-connection LSTM upgrades TD-LSTM to capture the connection between aspect and each context word.

ATAE-LSTM: Attention-based LSTM with Aspect Embedding joins (i) aspect attention vector with (ii) LSTM hidden vector for the sentiment polarity.

AF-LSTM: Aspect-fusion LSTM learns associative relationships between aspect and context words by word-aspect fusion attention.

HP-LTM: The authors introduced a hierarchical bidirectional strategy for extracting features.

MemNet: Deep memory networks captures the sentimental information without using a recurrent network. This model achieved significant improvements, but with more memory layers.

Skip-thoughts: It predicts the surrounding sentences, but its training is time-consuming.

Byte mLSTM: This model is one of the most remarkable in recent SC works, since it achieved superior results on customer reviews. It is able to predict the next character (byte) from preceding characters by using mLSTM [124] byte embedding.

VIRTUAL ADV: It is able to capture word-level information from unlabeled data and to learn the parameters. It achieves good performance for semi-supervised and supervised benchmark.

TopicRNN: It mixes RNN and topic modeling. It uses long-range dependencies to improve the output word probabilities of documents.

HN-ATT: It is a hierarchical network attention model for document classification. Additionally, it implements word-level and sentence-level by using two levels of attention mechanisms.

Structured Att: It backwardly infers passes for structured attention, and it uses edge marginal in structured models by the matrix-tree algorithm as attention weights.

3) RECURSIVE NEURAL NETWORK (RECNN)

RecNN is a generalization of RNN that applies recursively the same set of weights over a directed acyclic, but in a tree structure input (i.e., words and phrases in a hierarchical structure). RecNN models are linguistically motivated, as they explored tree structures (e.g., syntactic structures) and are aimed to learn complex compositional semantic. The tree structures used for RNNs include constituency tree and dependency tree. On one hand, in a constituency tree, the words are represented as leaf nodes, the phrases are represented as internal nodes and the root node represents the whole sentence. On the other hand, in a dependency tree, each node represents a word that is connected with other nodes with dependency connections. Generally speaking, in RecNN, the vector (parent) representation of each node is calculated from all of its children using a weight matrix. As shown in Figure 5 (center), RecNN recursively generates parent representations in a bottom-up fashion by combining tokens to produce representations (phrases), and eventually the whole sentence.

RecNN is used for different purposes. It has recently received attention for its compositionality in semantic vector spaces [42]; for instance, in [42], [43] the phrases and sentences are formed in a binary tree. Additionally, it has been used in successful prediction models in cutting-edge domains such as representing phrases. For instance, Qian *et al.* [108] encoded syntactic knowledge in the composition function of RNN. Dong *et al.* [51] transferred a dependency tree of a sentence into a target-specific recursive structure, and get higher level representation based on that structure. Researchers have mainly studied SC on RecNN, proposing models such as RAE [43], MVRNN [42], RNTN [36], DRNN [106] and CCAE [125].

RAE: Recursive Autoencoder automatically learns compositionality from sentences. This model could capture the meanings of texts by using predicting structures.

MVRNN: Matrix-vector RecNN represents the words as vectors to capture the meaning of long phrases.

RNTN: Recursive Neural Tensor Network are used for semantic compositionality of phrases.

DRNN: Deep RecNN are able to model sentences by using stacks of multiple recursive layers on top of each other.

CCAE: Combinatorial Category Autoencoders capture the compositional aspect of sentences by using combinatorial category grammar operators. It is used for multiple tasks without the need to re-train the main model.

4) HYBRID NEURAL NETWORK

The hybrid neural network uses more than one model in one task [122] and, usually, it achieves better accuracy in SC. Thus, several models have been used such as GrConv [126], LSTM-GRNN [25], Conv-GRNN [25] and Tree-LSTM [41]:

GrConv: Gated recursive CNN Joins a binary tree (using GRU) with CNN to automatically learn grammatical properties from a sentence.

Conv-GRNN and **LSTM-GRNN**: They share a CNN or LSTM with a GRNN to combine the sentence vectors to improve the classification on documents.

Tree-LSTM: Tree-LSTM learns representations using parsers of LSTM. However, it consumes a considerable training time because it needs to apply predefined syntactic structures.

IV. EXPERIMENTAL SETUP

This section reports the results of the previously mentioned DL experiments. The first two subsections (A and B) describe (i) the popular datasets used to train/test the models and (ii) the popular word embedding versions, respectively. The metrics used to evaluate the classification performance are described in Subsection C.

A. DATASETS

Table 3 summarizes the most commonly used datasets of reviews to evaluate SC approaches. It shows (i) the quantitative information (e.g., number of classes, size of the dataset, vocabulary size and length of reviews) and (ii) the qualitative information (e.g., level type, data source and review domain). As shown in the table, the datasets have different classes, sizes, domains, data extracted, labeled/unlabeled reviews, balanced/imbalanced reviews, rating variations of imbalanced, sentence's and token's lengths. These datasets are described below:

1) CUSTOMER REVIEWS (CR)

It is a widely used benchmark dataset[1] (extracted from Amazon) that includes 3,775 full-length customer reviews, where the sentences are labeled as positive or negative [50].

[1]These datasets are available at:
http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html

2) RT-2k AND RT-s

These two datasets[2] consist of movie reviews obtained from Rotten-Tomatoes (RT), where each customer review is classified as positive or negative. The main difference between both of them is in their size. On one hand, RT-s polarity dataset [10] contains 5,331 positive and 5,331 negative processed reviews of movies. On the other hand, RT-2k contains 1,000 positive snippets and 1,000 negative snippets.

3) STANFORD SENTIMENT TREEBANK (SST)

SST[3] is an extension of RT-s. It splits into training, development and testing. In SST, the sentences are parsed into parse trees. It includes two benchmark datasets (such as binary version and fine-grained) [36]. The binary version (*SST-2*) is labeled as positive/negative (no neutral reviews). Whereas in the fine-grained version (*SST-5*), reviews are classified as very positive, positive, neutral, negative and very negative.

4) IMDb-P AND IMDb-F

These datasets are large scale reviews of movies (full-length). Firstly, the Stanford polarity movie review dataset (IMDb-P)[4] [61] is equally divided between the training and the test reviews (one for positive and another one for negative). Secondly, *IMDb-Full* (*IMDb-F*) is split into ten sentiment labels as presented in [25]. The test set consists of 10% of the whole size.

5) YELP CHALLENGE[5]

Yelp dataset challenge [25] is obtained from Yelp.com and it contains restaurant reviews, which are organized or classified into five levels of rating. From this dataset, three subsets are constructed: *Yelp-13*, *Yelp-14* and *Yelp-15*. These subsets contain 335,018, 1,125,457 and 1,569,264 samples respectively. Each one of these three is split into (train/dev/test) sets with sizes (80/10/10), respectively.

6) Elec[6]

This dataset is a subset of the large Amazon dataset [127] that consists only of reviews of electronic products [72].

7) LARGE-SCALE DATASETS

These datasets[7] were obtained from Yelp.com and Amazon.com. *Yelp-P/F:* Two classification tasks (Full/Polarity) are constructed from this dataset. The first one predicts the number of stars the user has given, and the second one predicts a polarity label by considering stars 1 and 2 as negative, and 3 and 4 as positive. The full dataset has 130,000 training samples and 10,000 testing samples in each star, and the polarity dataset has 280,000 training

[2] https://www.cs.cornell.edu/people/pabo/movie-review-data/
[3]http://nlp.stanford.edu/sentiment/
[4]These datasets are available at:
http://ai.stanford.edu/amaas/data/sentiment
[4] https://www.yelp.com/dataset/challenge
[5] http://riejohnson.com/cnn_data.html
[7] https://github.com/zhangxiangxiao/Crepe

TABLE 3. Statistics of the 21 available review datasets after tokenization. Class: Number of target classes (opinion polarity). L: Average reviews' length. V: Vocabulary size. Test: Test set size (CV "Cross Validation" means there was no standard train/test split and thus 10-fold CV was used). Dist. (+, −): Lists the class distribution.

S#	Corpus		Class	Type	Size	L	Dist.(+,−)	Test	V	Data Source	Review Domain	Format Extracted
1	CR		2	Sentence-Level	3775	19	0.64/ 0.36 (Imbalance)	CV	6k	Amazon	Product	Each of sentence includes a text format and a title with tags of aspect terms
2	RT-2k		2		2000	786	0.5/0.5 (Balance)	CV	51k	Rotten-tomatoes	Movie	Each line in each text file corresponds to a single sentence (snippets)
3	RT-s		2		10662	20	0.5/0.5 (Balance)	CV	21k	Rotten-tomatoes	Movie	Version of SST-5, with neutral reviews removed and the remaining reviews categorized to either negative or positive.
4	Stanford Sentiment Treebank	SST-2	2		11855	18	(Unlabeled)	2210	18k	Rotten-tomatoes	Movie	For each example in the dataset, there exists only one sentence and a label associated with it. And the labels can be one of {negative, somewhat negative, neutral, somewhat positive, positive}.
5		SST-5	5		9613	19	(Unlabeled)	1821	16k	Rotten-tomatoes	Movie	
6	IMDb-P		2	Document-Level	50,000	230	0.5/0.5 (Balance)	25k	392k	IMDb	Movie	This dataset consist of informal reviews. It didn't allow no more than 30 reviews per movie.
7	IMDb-F		10		348,5k	326	07/.04/.05/.05/.08/.11/.15 /.17/.12/.18	34,85k	115k	IMDb	Movie	The reviews contain on user ratings (scaled from 0 to 10).
8	Yelp Challenge	Yelp-13	5		335k	152	.09/.09/.14/.33/.36 (Imbalance)	33.5k	211k	Yelp	Restaurant	each example consists of several review sentences and a rating score range from 1 to 5 (higher is better).
9		Yelp-14	5		1m	157	.10/.09/.15/.30/.36 (Imbalance)	100k	476k	Yelp	Restaurant	
10		Yelp-15	5		1.5m	152	.10/.09/.14/.30/.37 (Imbalance)	150k	613k	Yelp	Restaurant	
11	Elec		2		50,000	125	0.5/0.5 (Balance)	25k	40k	Amazon	Product	The data only includes the text section.
12	Large Scale	Yelp-P	2		598k	153	0.5/0.5 (Balance)	38k	116k	Yelp	Restaurant	Each sample is a piece of review text with a binary label (negative or positive).
13		Yelp-F	5		700k	155	0.2/0.2/0.2/0.2/0.2 (Balance)	50k	125k	Yelp	Restaurant	
14		Amazon-P	2		4m	93	0.5/0.5 (Balance)	400k	395k	Amazon	product	One review has a review title, a review content and a sentiment label.
15		Amazon-F	5		3.65m	91	0.2/0.2/0.2/0.2/0.2 (Balance)	650k	357k	Amazon	product	
16	SemEval	SE14-Lap	3	Aspect-Level	3845	--	0.61/0.18/0.21 (Imbalance)	800	--	Amazon	product	XML tag, in which two attributes ("from and "to") that indicate its start and end offset in the text.
17		SE14-Res	3		3841	--	0.45/0.21/0.34 (Imbalance)	800	--	Yelp	Restaurant	
18		SE15-Lap	3		2500	--	0.57/0.07/0.36 (Imbalance)	761	--	Amazon	Product	XML tag of {Entity # Attribute, polarity}.
19		SE15-Res	3		200	--	0.62/0.05/0.33 (Imbalance)	685	--	Yelp	Restaurant	XML tag of {Entity # Attribute, Opinion-Target-Expression, polarity}
20		SE16-Lap	3		3308	--	0.56/0.07/0.37 (Imbalance)	808	--	Amazon	Product	XML tag of {Entity # Attribute, polarity}.
21		SE16-Res	3		2676	--	0.66/0.04/0.30 (Imbalance)	676	--	Yelp	Restaurant	XML tag of {Entity # Attribute, Opinion-Target-Expression, polarity}.

TABLE 4. Description of the public released pre-trained word embedding datasets.

Learning Dense Embeddings	Embedding	Source	Training Corpus	Vocabulary Size	Tokens	Embedding's dimension	Web Resource (URL)	Training Time (epochs)
Neural Network	SENNA	[134]	Wikipedia	130,000	--	50	http://ronan.collobert.com/senna/	2 months (50)
	Turian	[139]	RCV1	268,810	1.8B	25, 50 or 100	--	few weeks (50)
	HLBL	[136]	Reuters	246,122	37M	50 or 100	http://metaoptimize.com/projects/wordreprs/	7 days (100)
	Word2Vec	[47]	Google News	3,000,000	1B	300	https://code.google.com/archive/p/word2vec/	--
		[66]	Amazon	1,000,000	4.7B	50 or 300	http://sentic.net/ AmazonWE.zip	--
	FastText	[137]	Facebook	--	--	--	https://github.com/facebookresearch/fastText	--
Matrix Factorization	Huang	[140]	Wikipedia	100,232	1.8B	50	http://ai.stanford.edu/ehhuang/	Weeks (50)
	GloVe	[138]	Wikipedia	400,000	--	50, 100, 200 or 300	http://nlp.stanford.edu/projects/glove/	--
		[138]	Twitter	1,200,000	--	25, 50, 100 or 200	--	--

samples and 19,000 test samples in each polarity, which are also obtained from [24]. *Amazon-P/F:* This product benchmark [24] has two versions (Full/Polarity) as in Yelp review dataset. The authors used the Stanford Network Analysis Project (SNAP) for obtaining on the reviews. The dataset contains 3,650,000 documents for a full dataset and a 4 million documents for a polarity dataset.

8) SEMANTIC EVALUATION (SemEval)

This dataset is created for the Aspect-Based Sentiment Analysis (ABSA) task with three polarities (positive, neutral or negative). It is a collection of review datasets (SE-14[8] [128], SE-15.[9] [129] and SE-16[10] [130]). These contain several domains and languages. The paper only focus on restaurant reviews and laptop reviews in English. For ABSA task, SE-14 is the most usually used because SE-15 and SE-16 datasets have conflict sentiment for each aspect based on its categories.

B. WORD EMBEDDING VERSIONS

Word embeddings are distributed representations of text that encode semantic and syntactic properties of words. They are usually represented as dense and low-dimensional vectors, by applying the previously mentioned approaches. In this section, we briefly discuss the available seven-word embedding datasets that are summarized in Table 4:

1) SENNA EMBEDDING

It is a semantic/syntactic extraction that uses a NN architecture [131]. SENNA behaves very fast and robust (it does not need parsed text) and it is able to label large and noisy corpora. It performs composition over the learned word vectors for classification. In [132], SENNA achieved good results compared with other embedding approaches such as Word2Vec and GloVe.

2) TURIAN EMBEDDING

This uses word embedding in semi-supervised learning. This embedding covers 268,810 words, each one is represented by 25, 50 or 100 dimensions. The weakness of this is that it cannot fully exploit the potential of word vectors.

3) HLBL EMBEDDING

It is a hierarchical log-bilinear model presented by [133]. This embedding covers 246,122 words (one year of Reuters English newswire from August 1996 to August 1997).

4) Word2Vec EMBEDDING

It is an unsupervised learning tool that is provided with two architectures for computing vector representations of words: the continuous bag-of-words and skip-gram. The first one predicts the target word from its context words, while the second does the inverse. Consequently, the skip-gram is better for larger dataset. The phrases were obtained using a simple data-driven approach described in [19]. It is characterized by its speed even with a huge dataset. However, it does not take into account the linguistics of words.

5) FastText EMBEDDING

It is a skip-gram with sub-word character n-grams [134]. It is similar to word2vec and faster for training and evaluation.

[8] http://alt.qcri.org/semeval2014/task4/
[9] http://alt.qcri.org/semeval2015/task12/
[10] http://alt.qcri.org/semeval2016/task5/

6) HUANG EMBEDDING

It requires context to disambiguate words. Therefore, Huang incorporates global context to deal with challenges raised by words with multiple meanings. Its size is 100,232 word embedding. The training corpus is: April 2010 snapshot of Wikipedia.

7) GloVe EMBEDDING

It is a global vector that has trained on the nonzero entries of a global word-word co-occurrence matrix [135].

C. PERFORMANCE APPRAISAL

The accuracy (Acc) metric (calculated using Equation 1) [138] is commonly used to measure the performance of SC approaches ([13], [27], [139], [139], [140]). It refers to the proportion of correctly classified samples over the whole samples. Acc is calculated through a confusion matrix. For instance, Table 5 shows the confusion matrix for two classes (Positive, Negative). TP (True Positive) means a positive observation which is predicted as positive, FN (False Negative) means a positive one which is predicted as negative, TN (True Negative) means a negative one which is predicted as negative, and finally FP (False Positive) means the observation is negative and is predicted as positive.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

TABLE 5. The confusion matrix.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Although Acc has been chosen by most of the researchers to measure the performance of DL-based SC approaches, other metrics such as precision, recall and f-measure could provide better insights. These other metrics are not used in DL-based SC approaches, although some researchers have used them in other SA tasks such as aspect extraction ([27], [139]). Additionally, other interesting metrics, such as weighted kappa [141], are used to distinguish between disagreements (misclassification) of negative to positive. A study by Novielli *et al.* [142] analyzed misclassification between sentiment tools in software engineering and concluded that even though there are disagreements among the tools, it is minimal and dataset dependent. However, weighted kappa is not used in DL-based sentiment classification approaches. Consequently, since our comparison is a literature-based one, we were forced to only use the Acc metric to compare the different DL-based SC approaches. Researchers in this field could be motivated to use and address other metrics in their experiments in order to have the possibility to apply other metrics for broader comparisons.

V. A COMPARISON OF DL-BASED SC APPROACHES

In this section, we compare the performance of SC techniques in three domains: product, movie and restaurant reviews. As it was previously mentioned, these three domains entail interesting challenges and constitute the most common domains in review mining. The performance of the state-of-the-art approaches on the available datasets are summarized in Tables 6-8. The provided results in these tables are reproduced under the same measure (accuracy) and conditions that they were stated in the original papers.

A. PRODUCT REVIEWS

The analysis of product reviews is very important for manufacturers/companies to understand the customer opinions as well as for the customers to know other consumers' feedbacks. Unfortunately, the extracted reviews from this domain contain a considerable amount of noisy data. So, a huge effort in preprocessing the datasets is needed. The results of applying SC approaches on the seven datasets (CR, Amazon-P, Amazon-F, Elec, SE14-Lap, SE15-Lap and SE16-Lap) are summarized in Table 6. For each data set, the accuracy resulted from each approach is added to the table. As shown in Table 6, Byte mLSTM [32] model significantly outperformed other methods on CR dataset, while TreeNet achieved better performance and generalization with fewer parameters. In Byte mLSTM, a byte-level language model trained on the large product review dataset is used to obtain sentence representations. For the Elec datasets, ADV-LM-LSTM+IMN [143] gave the best result, where ADV-LM-LSTM [65] provides a performance competitive with the current best result for the configuration of supervised learning. In [143], they added the IMN to improve the performance. VAT-LM-LSTM [88] has lower performance than VIRTUAL ADV [65] on Elec, while VAT-LM-LSTM is the most reliable result for the comparison as expected by [143]. In [65], they applied perturbations to the initial pre-trained word embeddings in conventional LSTM. On Amazon (polarity/full), DPCNN [152] achieved the highest accuracy, although it gave lower performance in other domains (e.g., the restaurant domain shown in Table 8). The DPCNN model develops CNN for capturing the n-gram features (i.e., can capture n-gram information of different sizes without manually setting the convolution kernel size). On SE14-Lap, BBLSTM-SL model [151] achieved the second best reported result, only after the significantly LCR-Rot [146] model, where this model is able to represent the sentiment aspect better, especially when the aspect contains multiple words. On SemEval 15/16 datasets of laptop domain, the system that is presented by [153] is competitive. This system is scalable and able to process a high volume of opinion-based documents in real-time.

TABLE 6. Comparison with state-of-the-art results from the literature on the most widespread available Product review datasets on SC.

DS	Model	Acc	DS	Model	Acc	DS	Model	Acc	DS	Model	Acc
CR	DisSent [96]	84.9	Amazon-P	Char-CRNN [55]	94.10	SE14-Lap	PF-CNN [83]	70.06	SE16-Lap	Senti [133]	74.28
	3W-CNN [77]	85.8		FastText [137]	94.60		GRNN-G3 [100]	71.47		[147]	74.58
	CNN-multichannel [40]	85.0		BiLSTM + KNN [148]	94.70		Feature-enhanced SVM [149]	72.10		LeeHu [133]	75.91
	CNN-Rule-q [120]	85.3		Char-level CNN [24]	95.07		IAN [101]	72.10		AUEB [133]	76.9
	QuickThoughts [98]	86.0		BiLSTM + KNN [148]	95.30		MemNet [35]	72.37		NileT [133]	77.40
	AdaSent [30]	86.3		Region-embedding [150]	95.30		Coattention-MemNet [143]	72.90		IHS-R [133]	77.90
	SuBiLSTM [151]	86.5		SANet [84]	95.48		DMN [102]	72.95		ECNU [152]	78.15
	MultiTask [97]	87.7		VDCNN [153]	95.69		Coattention-LSTM [143]	73.50		IIT-T [133]	78.40
	TreeNet [113]	88.4		word-CNN [114]	96.21		BBLSTM-SL [154]	74.90		INSIG [133]	78.40
	Byte mLSTM [32]	91.4		DPCNN [155]	96.68		LCR-Rot [149]	75.24		[156]	81.08
Elec	oh-2LSTMp [157]	93.92	Amazon-F	Char-CRNN [55]	59.20	SE15-Lap	wnlp [132]	72.07			
	One-hot CNN [157]	94.13		Char-level CNN [24]	59.57		EliXa [132]	72.91			
	iAdvT-Text [91]	94.42		FastText [137]	60.20		TJUdeM [132]	73.23			
	One-hot bi-LSTM [157]	94.45		BiLSTM [148]	60.30		LT3 [132]	73.76			
	VAT-LM-LSTM [68]	94.46		Region-embedding [150]	60.80		[50]	75.87			
	LM-LSTM+IMN [146]	94.52		SANet [84]	61.33		[147]	76.54			
	VIRTUAL ADV [68]	94.60		VDCNN [153]	63.00		Lsislif [132]	77.87			
	LML [90]	94.76		DCCNN-ATT [5]	63.00		ECNU [152]	78.29			
	iVAT-LSTM [91]	94.82		word-CNN [114]	63.76		sentiue [158]	79.34			
	ADV-LM-LSTM+IMN [146]	94.86		DPCNN [155]	65.19		[156]	85.89			

B. MOVIE REVIEWS

Movie reviews constitute the most challenging domain within review mining. Data is usually extracted from Rotten-tomatoes and the Internet Movie Database (IMDb). The extracted reviews do not offer a clear idea of what are factual information and which are mainly opinions. Thus, movie reviews are apparently harder to classify than product reviews, since product reviews have less specific features [18], [52]. This is why the movie domain is experimentally convenient and has larger online reviews [17]. Table 7 gathers a list of approaches and their results on six public movie review datasets. As shown in this table, DAN [105] model has achieved the worst performance on most of the datasets because it did not initialize with pre-trained embedding or randomly. The Byte mLSTM model is competitive with more complex models on the SST-2 and it outperforms the state-of-the-art models. But with SST-5 and IMDb-P datasets, it did not perform well. This model trained a simple logistic regression classifier with L1 regularization. It achieved the second best result on the RT-2k dataset after AC-BiLSTM model. On RT-2K, AC-BiLSTM [156] achieved the highest accuracy on LSTM based models and entails an improvement of 10.8, 9.7, 9.2 and 6.1 over TE-LSTM, SA-LSTM, SATA TLSTM, and Byte mLSTM, respectively. AC-BiLSTM model is based on a bidirectional LSTM and attention mechanism. Bidirectional LSTM is adopted to access the preceding and succeeding context representations. While the attention mechanism is intended to give focus on the information produced by the hidden layers of BiLSTM. CRAN [112] gives best results on RT-S dataset, where it combines RNN with the convolutional attention model to resolve aspect-level sentiment analysis tasks as well. The BCN ELMo [157] method achieved better results than other methods on SST-5. These methods combined the Biattentive Classification Network (BCN) with Embeddings from Language Models (ELMo) word representations to encode sentences to pass through classifiers. TCV [158] in movie reviews outperforms the other models, unlike restaurant reviews. This model proposed Text Concept Vector (TCV) for the text representation which extracts the concept level information of text. At the document-level IMDb-P task, the LML [87] model is competitive with more complex models. This model applied entropy minimization to unlabeled data in an unsupervised way. Thus, it outperforms VIRTUAL ADV and iVAT-LSTM. H-CRAN [112] achieved the second best result on the IMDb-F dataset after TCV model. TCV model obtains a substantial improvement in classification accuracy compared with the more complex methods, for the reason given above.

C. RESTAURANT REVIEWS

Table 8 gives the results of different SC models on different restaurant datasets and it can be perceived that H-CRAN model achieved good performance on the three yelp datasets (Yelp13, Yelp-14 and Yelp-15). For these three datasets, H-CRAN [112] outperforms LSTM-GRNN [25] model by 3.6%, 4.4% and 5.4%, respectively. H-CRAN used a similar hierarchical architecture than LSTM-GRNN, but with an additional attention mechanism to extract salient words in sentences and salient sentences in the document.

TABLE 7. Comparison with state-of-the-art results from the literature on the most widespread available *Movie* review datasets on SC.

DS	Model	Acc	DS	Model	Acc	DS	Model	Acc
RT-2k	DAN [108]	80.3	SST-2	MVCNN [118]	89.4	IMDb-P	RCRN [164]	92.8
	CNN-Rule-q [120]	81.7		TE-LSTM [94]	89.6		Byte mLSTM [32]	92.88
	TE-LSTM [94]	82.2		NSE [165]	89.7		CNN+tvEmb [166]	93.49
	QuickThoughts [98]	82.4		IRAM [116]	90.1		Ensemble [167]	93.51
	MultiTask [97]	82.5		CT-LSTM [168]	90.2		TopicRNN [33]	93.72
	SA-LSTM [169]	83.3		BCN [164]	90.3		ADV-LM-LSTM+IMN [146]	93.93
	VIRTUAL ADV [68]	83.4		AR-Tree [164]	90.4		oh-2LSTMp [157]	94.06
	SATA TLSTM [170]	83.8		RCRN [171]	90.6		VIRTUAL ADV [68]	94.09
	Byte mLSTM [32]	86.9		SATA TLSTM [170]	91.3		iVAT-LSTM [91]	94.34
	AC-BiLSTM [159]	93.0		Byte mLSTM [32]	91.8		LML [90]	95.68
RT-s	CNN [40]	81.5	SST-5	TE-LSTM [94]	52.6	IMDb-F	SVM + TextFeatures [25]	40.5
	RR-CNN [172]	81.6		Byte mLSTM [32]	52.9		CNN [40]	40.6
	SA-SNN [44]	82.1		NTI [173]	53.1		SVM + Bigrams [25]	40.9
	HS-LSTM [174]	82.1		CT-LSTM [168]	53.6		Conv-GRNN [25]	42.5
	SFCNN [175]	82.7		AR-Tree [164]	53.7		FastText [137]	45.2
	AdaSent [30]	83.1		IRAM [116]	53.7		LSTM-GRNN [25]	45.3
	AC-BiLSTM [159]	83.2		Gumbel Tree-LSTM [117]	53.7		Structured Att [126]	49.2
	ACNN [176]	83.4		RCRN [164]	54.3		HN-ATT [87]	49.4
	TreeNet [113]	83.6		SATA TLSTM [170]	54.4		H-CRAN [115]	50.2
	CRAN [115]	83.8		BCN ELMo [160]	54.7		TCV [161]	50.5

TABLE 8. Comparison with state-of-the-art results from the literature on the most widespread available *Restaurant* review datasets on SC.

DS	Model	Acc	DS	Model	Acc	DS	Model	Acc	DS	Model	Acc
Yelp-13	CNN [40]	62.7	Yelp-15	Paragraph Vector [47]	60.5	Yelp-F	SWEM [177]	63.79	SE15-Res	EliXa [132]	70.05
	Conv-GRNN [25]	63.7		SVM + TextFeatures [25]	62.4		FastText [137]	63.90		UMDuluth [132]	71.12
	FastText [137]	64.2		SVM + Bigrams [25]	62.4		SANet [84]	63.97		SIEL [132]	71.24
	LSTM-GRNN [25]	65.1		CNN [40]	64.5		VDCNN [153]	64.72		wnlp [132]	71.36
	TCV [161]	67.8		Conv-GRNN [25]	66.0		Region-embedding [150]	64.90		UFRGS [132]	71.71
	Structured Att (doc) [126]	67.8		FastText [137]	66.6		[178]	65.83		LT3 [132]	75.02
	Structured Att [126]	68.0		LSTM-GRNN [25]	67.6		DCCNN-ATT [5]	66.00		lsislif [132]	75.50
	HN-ATT [87]	68.2		HN-ATT [87]	71.0		word-CNN [114]	67.61		[156]	77.94
	Structured Att (both) [126]	68.6		TCV [161]	71.5		DPCNN [155]	69.42		ECNU [152]	78.10
	H-CRAN [115]	68.7		H-CRAN [115]	73.0		ULMFiT [162]	70.02		sentiue [158]	78.69
Yelp-14	Paragraph Vector [47]	59.2	Yelp-P	FastText [137]	95.70	SE14-Res	Coattention-LSTM [143]	78.80	SE16-Res	[147]	81.62
	CNN [40]	61.4		VDCNN [153]	95.72		PG-CNN [83]	78.93		INSIG [133]	82.07
	SVM + Bigrams [25]	61.6		SWEM [177]	95.81		PF-CNN [83]	79.20		[50]	83.18
	SVM + TextFeatures [25]	61.8		DCNN [179]	96.04		GRNN-G3 [100]	79.55		AUEB [133]	83.24
	Conv-GRNN [25]	65.5		Region-embedding [150]	96.40		Coattention-MemNet [143]	79.70		ECNU [152]	83.59
	FastText [137]	66.2		DCCNN-ATT [5]	96.50		Feature-enhanced SVM [149]	80.89		IHS-R [133]	83.94
	LSTM-GRNN [25]	67.1		[178]	96.58		MemNet [35]	80.95		NileT [133]	85.45
	TCV [161]	69.2		word-CNN [114]	97.10		BBLSTM-SL [154]	81.30		IIT-T [133]	86.73
	HN-ATT [87]	70.5		DPCNN [155]	97.36		LCR-Rot [149]	81.34		[156]	87.10
	H-CRAN [115]	71.5		ULMFiT [162]	97.84		DMN [102]	81.43		XRCE [163]	88.13

ULMFiT [159] model outperforms other models on the large scale datasets Yelp-P and Yelp-F. ULMFiT model is well known for their transfer learning capabilities and could be generalized for various NLP tasks across different domains. On SE14-Res, DMN [99] implemented multiple attention mechanism on LSTM and it outperforms all others.

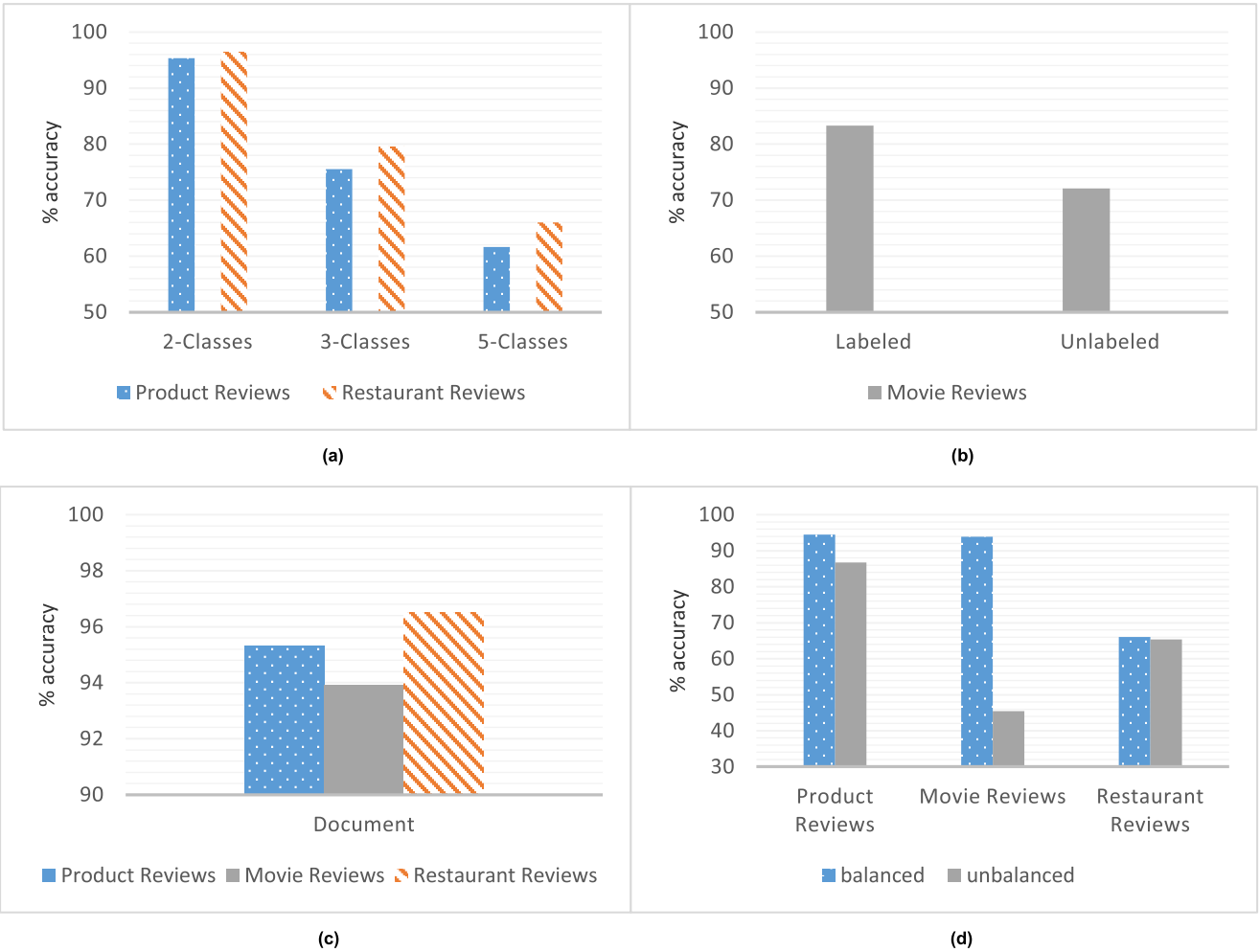

FIGURE 6. Comparison of SC approaches on the review domains based on the *data preparation* based factors.

FastText [134] model is widespread on document-level SC of this domain. FastText has been shown to be trained quickly and it achieves high prediction performance comparable to RNN embedding model for SC. Compared to FastText, many models gave superior performance across the board. ECNU (..) [149] model achieved the second best reported result on the SE15-Res dataset, only after the significantly slower Sentiue [155] model. It is also competitive on SemEval 15/16 datasets. ECNU used a lexicon-based approach based on domain-specific lexicons, while Sentiue used a MaxEnt classifier with a set of features including lexicons. On SE16-Res, XRCE [160] model outperforms all other models. This model used symbolic parser designed with special lexicon and combined with SVM.

Furthermore, we have compared the addressed DL-based SC approaches on the three domains products, movies and restaurants reviews; by using the three classes of factors suggested in Section III. The average of the approaches performances is used on the three domains for the different factors. These factors-based comparisons are discussed in the following subsections.

1) DATA PREPARATION BASED COMPARISON

The comparison of the factors in data preparation phase on the three review domains shows the average performance of SC approaches for each factor. As shown in Figure 6, the factors magnitudes, annotation, granularity and equilibrium are able to change performance of SC as follows. First, the results show that the more number of classes in a review-dataset, the lower performance of the applied models, as shown in Figure 6(a). This comparison is based on the Large-Scale dataset from 2 and 5 classes, while in 3-classes, the comparison is based on the SemEval datasets. Also, a slight superiority of restaurant-reviews over product-reviews is noted. The movie-reviews datasets are not used as no ternary classes are available for this domain. Second, in Figure 6(b), using Rottentomatoes movie reviews to compare both labeled (e.g., RT-2 and RT-S) and unlabeled data (e.g., SST-2 and SST-5), the comparison shows that labeled-data dominate unlabeled-data on 2 and 5 classes. Third, these domains are compared on document granularity level. Document-level achieves the highest performance from others in the same class (e.g., binary class) as in Yelp-p, Amazon-p, Elec,

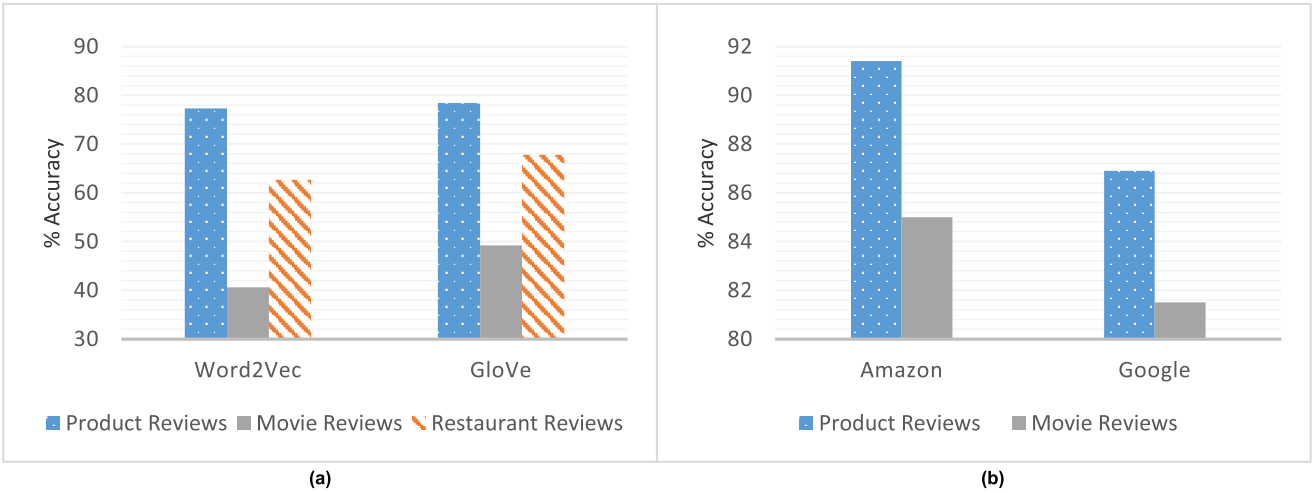

FIGURE 7. Comparison of SC approached on the review domains based on *word embedding*.

IMDb-p, SST-2 and CR. In Figure 6(c), three datasets from the binary class are compared at document-level. In this level, restaurant-reviews beat the others as they have less noise data. Fourth, movie-review datasets (e.g., IMDb-F and IMDb-P) have high performance on balanced-data. This may happened due to the big difference among the number of classes they have. So, product reviews are compared at the same class but not the same level (e.g., CR and Elec). Also, balanced-data is outdone. On restaurant reviews, the same class level and the same size approximately (e.g., Yelp-14 and Yelp-F) are compared. The results showed that the balanced-data outperformed, as shown in Figure 6(d).

2) FEATURE REPRESENTATION BASED COMPARISON

This section compares two factors in the feature representation phase: Learning word embedding and training corpus type. At the end of this comparison, we will be able to answer the following two questions: "*What is the most appropriate way to learn dense embedding in SC task?*" and *"What is the best data type for training in SC task?"* For the first factor, we compare *Word2Vec* of neural network and *GloVe* of matrix factorization that are the most prevalent (due to their high performance on SC) on review domains. Figure 7(a) offers a comparison on the 3 domains in document-level on different classes (e.g., IMDb-F, Amazon-P/F and Yelp-13). The figure shows that matrix factorization-based method (i.e., GloVe) is superior to the other. In the second factor, we provide a comparison on Word2Vec for the two corpora Amazon and Google in movie-reviews and product-reviews domains (as Word2Vec is more prevalent in both domains). As shown in Figure 7(b), superiority of Amazon corpus-based methods is found in the two review domains (products and movies) over Google corpus-based methods. This happens due to the big opinionated information in Amazon corpus.

3) TECHNIQUE BASED COMPARISON

This section compares SC approaches based on the two factors: determining the level of the granularity and the architecture of the DL model. Figure 8 shows that the two factors affect the performance of SC. In the first factor, we compare the results of the DL-methods from different review domains at the 3 levels: document-level, sentence-level and aspect-level, as shown in Figure 8(a). We did not find a DL method that has been applied to the three levels at the same time. Therefore, we compare the highest average performance of the datasets at each level (i.e., using Yelp-P/ Amazon-P of document-level, SST-2/CR of sentence-level and SE16-Res/ SE16-Lap of aspect-level). We have noticed that document-level has the best performance as compared to the others. Therefore, in the second factor, we concerned in comparing DL architectures (e.g., CNN, RNN, RecNN and Hybrid) on this level, as shown Figure 8(b). We have found that RNN is better than the others. In this comparison, RecNN methods are excluded because these methods are rarely applied (for its lack of quality at this level). In [152], they tried on the level of documents in movie and product reviews such as IMDb-P and Elec, respectively, in binary class. We also find a clear advantage of RNN over CNN and hybrid.

4) ANALYSIS OF THE RESULTS

The results of SC approaches on the three different domains are summarized in Tables 6-8. This section analyzes the most important and recent DL-methods at the three different granularity levels: aspect, sentence and document:

At the **aspect-level** SC, many researchers use RNN (e.g., GRNN-G3 [97], MemNet [35], IAN [98], Coattention-LSTM/MemNet [140] and BBLSTM-SL [151]). These approaches were applied on SemEval-14 (Restaurant and Laptop datasets) for the ABSA task. Recently, Coattention-LSTM/MemNet used location information. These are more effective than MemNet and IAN over Laptop datasets. But, in restaurant datasets, MemNet outperforms both Coattention-LSTM/MemNet and IAN. MemNet model did not depend on sentiment lexicon or syntactic parser, and it is computationally efficient. BBLSTM-SL model is more

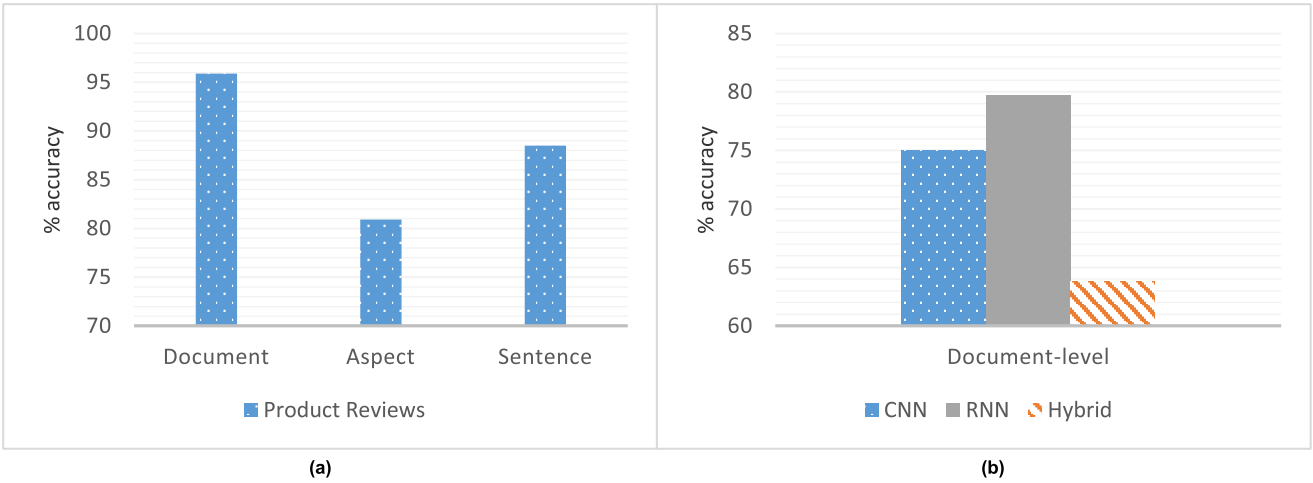

FIGURE 8. Comparison of SC approached on the review domains based on the used *DL techniques*.

effective than the previous models, since it utilized sentiment lexicon as a feature for words. Also, CNN has a bit of luck for aspect-based SC task (e.g., PG-CNN [80], PF-CNN [80]). The results of these approaches have the worst performance as compared to RNN approaches.

At **sentence-level** SC, all DL models have been used widely. There are two subtasks on this level (subjectivity classification and sentence-level SC), but we only focused on the second one which is the most common and used with all NN architectures: CNN approaches (CNN-Rule-q [117]), RNN approaches (QuickThoughts [95], MultiTask [94], Byte mLSTM [32]) and RecNN approaches (AdaSent [30], TreeNet [110]). It is interesting to say that CNN-Rule-q method is learned simultaneously during the training from labeled instances (comparing product and movie reviews of the three architectures). However, it has the worst performance as compared to the others. In RecNN, TreeNet is more effective than AdaSent on the two domains. The RNN approaches are also widespread in this level of SC. The Byte mLSTM model is applied in the document-level, but it is more robust in sentence-level (especially binary corpus). It achieves better results than QuickThoughts and MultiTask on the two domains (Products and Movies).

At **document-level** SC, both RNN and CNN are competitors, although RNN outperforms CNN. The comparison on the Product and Restaurant datasets show that DPCNN [152] (a CNN method) consistently outperforms VDCNN [150] and Hybrid approaches such as (DCCNN-ATT [5] and word-CNN [152]). The RNN model is common on the Product and Movie datasets together (e.g., oh-2LSTMp [154], LML [87], iAdvT-Text [88], LM-LSTM+IMN [143]). The most robust in performance from these approaches is LML on Movie reviews while LM-LSTM+IMN [143] on Product reviews. Also, the RNN is publically on the implementations of Movie and Restaurant reviews together such as HN-ATT [84], Structured Attention [123]. HN-ATT [84] achieves better results than the other method on Movie datasets. We have concluded from this analysis that RNN also excels at the aspect level and sentence level over the others.

VI. DISCUSSION AND OPEN ISSUES

This section presents extra factors that have not been previously mentioned and that are expected to affect the performance of SC approaches. Finally, some open issues in SC are introduced.

There are general factors in **data preparation** as domain determination, underlying language and corpus type. In *domain determination*, there are many domains on SA field: blogs, reviews (hotels, electronics, restaurants, etc.), social media and others. In our work, we focused on the review domains since it is the most prevalent in SC. In the selected field, there are also some aspects to take into account, for instance, reviews are not easily processed because they may contain irony or slangs. Moreover, the *underlying language* is an important factor for changing the performance in the SC task. For some languages, like English, there are many tools that enhance the performance of the classification problem. Sometimes, a model gives good results with a language, but not so well with others. For instance, A Multinomial Naïve Bayes model gave the best results for English, Support Vector Machine model for Dutch, and Maximum Entropy model for French as reported in [3]. We have used English because it is a widespread language.

Besides, the factors presented in Table 3 affect the performance of the DL-based SC approaches. That is, corpus's size, data source (Is the data extracted from different websites similar?), CV-based split (Some of the experiments were based on cross validation, others were divided into specific proportions), review length and vocabulary size, and the format of the data extracted (some of them were text and others were xml) have an important influence in the analysis.

A. CORPUS'S SIZE

As noted in the results, large scale datasets (e.g., Yelp-P and Amazon-P) achieved better results than smaller ones (e.g., CR and RT.2k). Corpus size and corpus domain have an effect on the system performance [16].

B. DATA SOURCE

We focused on corpora that were built on a rating system such as Amazon, Yelp, IMDb and rottentomatoes because they are adequate for automatic labeling and professional review sites.

C. CROSS-VALIDATION-BASED SPLIT

Some of the experiments were based on CV, others were divided into specific proportions. CV is used to estimate the predictive performance of the models more accurate than dividing data by specific proportions (e.g., 20% test-set and 80% train-set). Therefore, we have noted that the results with cross-validation are relatively less. For instance, SST-2 and RT-s are nearly identical, but SST-2 is more accurate than RT-s with cross-validation.

D. REVIEW'S LENGTH AND VOCABULARY SIZE

We have noticed that RT.2k achieved higher performance than RT.s, although the size of corpus is less. This is because the length of the reviews and variety of vocabulary in RT.2k are higher than in RT.s.

E. FORMAT EXTRACTION

Reviews have different formats. Some reviews are text and others are XML. Consequently, we have noticed that results of the two format are not the same.

Furthermore, there are other three key factors on feature representation: training corpus size, accuracy of embedding and method type. For the first factor, the increase of corpus size positively affects the performance of SC. Thus, word embedding versions must be trained with large data ([177], [67]). Even with similar domain, a huge corpus proved its effectiveness [175]. Severyn and Moschitti [178] trained word embedding using Word2Vec on 50 million tweets. Word2Vec method is learned on 408 million words of Wikipedia in [177]. For the second factor, some algorithms have been proposed to increase the accuracy of pre-trained word embedding [178], [179], as the amendment increases the accuracy in SC. For the third factor, method type, there are three types of feature representation affect the performance of DL-based methods: supervised, semi-supervised and unsupervised. As for SC, there are several DL models that have been applied to the three classifiers of feature representation, as shown in Table 9. First, supervised methods require more labeled data to work well. Second, semi-supervised methods use large unlabeled data with small labeled data to reduce sparse data. Their approaches improve generalization accuracy, but it consumes time to adapt with supervised systems. Third, unsupervised methods are trained on unlabeled data separately, but it is not sufficient for SC [126]. It used to improve classification accuracy [88]. We think these factors are effective, so they need further discussion from the researchers in this domain.

TABLE 9. Allocations of literatures based on SC with the three types of feature representations.

Method Type	*Articles' References*
Supervised	[129], [125], [30], [36], [41], [109], [43], [182], [183], [128], [42], [151]
semi-supervised	[166], [169], [68], [90], [146]
Unsupervised	[155], [184], [47], [38], [108], [75], [32], [185], [186], [24], [137], [98]

The proposed factors that affect the performance of SC approaches give researchers a whole insight to enhance the performance of their SC approaches. It also highlights three open issues that could be listed as follows.

- Reviews collected from various resources may contain irony, colloquial language, abbreviation, slangs, noisy, wrongly spelt or non-grammatical text. There is no automatic system for solving these gabs. So, the data preparation phase consumes the maximum time to convert data into a structured format.
- There is a lack of SC tools in non-English languages. For instance, Arabic has no enough tools, besides its parser complexity and synonymous dis-ambiguity make the classification problem a challenge. Thus, large companies suffer from this problem when analyzing customers' opinions of different languages. Furthermore, there is no system that works equally well on various review domains, due to the different orientation of opinion words for each field.
- Few attempts were made to create word embedding that combined the advantages of both neural network and matrix factorization, and increase the accuracy of embedding.

VII. CONCLUSION

This paper is a comprehensive survey of the state-of-the-art Deep Learning (DL) methods for on Sentiment Classification. Our results showed that although DL models give high performance and outperform others, many factors could affect the performance of these DL-based SC approaches. Besides, we have provided different literature-based comparisons to find the most significant factors in the three phases: data preparation, feature representation and classification techniques. The comparisons are conducted using more than 100 models and applied to 21 textual datasets.

In the *data preparation* based comparisons, we have obtained that: (i) The greater number of classes in a dataset, the more challenging SC problem; (ii) Labeled-datasets

achieve better results than unlabeled-datasets; (iii) When comparing different review domains (products, movies and restaurants) on document level, it is found the superiority of the restaurant reviews while the movie reviews have the least accuracies; and (iv) Balanced datasets outperformed imbalanced-datasets. In the feature representation based comparisons, we have deduced that: (i) Word embedding based matrix factorization (i.e., GloVe) is better than word embedding based neural network in the performance for SC (i.e., Word2Vec); and (ii) Training data on Amazon corpus with Word2Vec dense embedding is more appropriate for SC than Google corpus. In the classification technique based comparison, we have found that: (i) the three most common NN architectures (RNN, CNN and Hybrid NN) are widely used to solve the three levels (document, sentence and aspect) of SC, and they perform well with the document-level SC. RNN achieved the highest results as compared to the others; and (ii) the performance of DL-based models on the aspect-level SC is still a challenge and needs much more effort to be solved.

In this survey, we only focused on comparing results of DL models on benchmark datasets of customers' reviews. In the future, we hope to compare DL models on different word embedding versions in-depth. We expect that these kinds of in-depth comparisons and analysis provide researchers in this field by more factors that affect the performance of SC.

REFERENCES

[1] X. Zheng, Z. Lin, X. Wang, K.-J. Lin, and M. Song, "Incorporating appraisal expression patterns into topic modeling for aspect and sentiment word identification," *Knowl.-Based Syst.*, vol. 61, pp. 29–47, May 2014, doi: 10.1016/j.knosys.2014.02.003.

[2] K. Zhang, Y. Xie, Y. Yang, A. Sun, H. Liu, and A. Choudhary, "Incorporating conditional random fields and active learning to improve sentiment identification," *Neural Netw.*, vol. 58, pp. 60–67, Oct. 2014, doi: 10.1016/j.neunet.2014.04.005.

[3] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual Web texts," *Inf. Retr.*, vol. 12, no. 5, pp. 526–558, Oct. 2009, doi: 10.1007/s10791-008-9070-z.

[4] L.-S. Chen, C.-H. Liu, and H.-J. Chiu, "A neural network based approach for sentiment classification in the blogosphere," *J. Informetrics*, vol. 5, no. 2, pp. 313–322, Apr. 2011, doi: 10.1016/j.joi.2011.01.003.

[5] S. Wang, M. Huang, and Z. Deng, "Densely connected CNN with multi-scale feature attention for text classification," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4468–4474, doi: 10.24963/ijcai.2018/621.

[6] A. Reyes and P. Rosso, "Making objective decisions from subjective data: Detecting irony in customer reviews," *Decis. Support Syst.*, vol. 53, no. 4, pp. 754–760, Nov. 2012, doi: 10.1016/j.dss.2012.05.027.

[7] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowl.-Based Syst.*, vol. 89, pp. 14–46, Nov. 2015, doi: 10.1016/j.knosys.2015.06.015.

[8] C. Quan and F. Ren, "Unsupervised product feature extraction for feature-oriented opinion determination," *Inf. Sci.*, vol. 272, pp. 16–28, Jul. 2014, doi: 10.1016/j.ins.2014.02.063.

[9] L. Garcia-Moya, H. Anaya-Sanchez, and R. Berlanga-Llavori, "Retrieving product features and opinions from customer reviews," *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 19–27, May 2013, doi: 10.1109/MIS.2013.37.

[10] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2004, p. 271.

[11] A. Weichselbraun, S. Gindl, and A. Scharl, "Enriching semantic knowledge bases for opinion mining in big data applications," *Knowl.-Based Syst.*, vol. 69, pp. 78–85, Oct. 2014, doi: 10.1016/j.knosys.2014.04.039.

[12] A. Weichselbraun, S. Gindl, and A. Scharl, "Extracting and grounding contextualized sentiment lexicons," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 39–46, Mar. 2013, doi: 10.1109/MIS.2013.41.

[13] D. Bollegala, D. Weir, and J. Carroll, "Cross-domain sentiment classification using a sentiment sensitive thesaurus," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1719–1731, Aug. 2013, doi: 10.1109/TKDE.2012.103.

[14] Y. Seki, N. Kando, and M. Aono, "Multilingual opinion holder identification using author and authority viewpoints," *Inf. Process. Manage.*, vol. 45, no. 2, pp. 189–199, Mar. 2009, doi: 10.1016/j.ipm.2008.11.004.

[15] M.-T. Martín-Valdivia, E. Martínez-Cámara, J.-M. Perea-Ortega, and L. A. Ureña-López, "Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches," *Expert Syst. Appl.*, vol. 40, no. 10, pp. 3934–3942, Aug. 2013, doi: 10.1016/j.eswa.2012.12.084.

[16] M. R. Saleh, M. T. Martín-Valdivia, A. Montejo-Ráez, and L. A. Ureña-López, "Experiments with SVM to classify opinions in different domains," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14799–14804, Nov. 2011, doi: 10.1016/j.eswa.2011.05.070.

[17] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. ACL-Conf. Empirical Methods Nat. Lang. Process.*, vol. 10, 2002, pp. 79–86.

[18] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 417–424.

[19] Z.-H. Deng, K.-H. Luo, and H.-L. Yu, "A study of supervised term weighting scheme for sentiment analysis," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3506–3513, Jun. 2014, doi: 10.1016/j.eswa.2013.10.056.

[20] Z. Hailong, G. Wenyan, and J. Bo, "Machine learning and lexicon based methods for sentiment classification: A survey," in *Proc. 11th Web Inf. Syst. Appl. Conf.*, Sep. 2014, pp. 262–265.

[21] H. Cho, S. Kim, J. Lee, and J.-S. Lee, "Data-driven integration of multiple sentiment dictionaries for lexicon-based sentiment classification of product reviews," *Knowl.-Based Syst.*, vol. 71, pp. 61–71, Nov. 2014.

[22] 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, doi: 10.1109/MIS.2009.105.

[23] R. Moraes, J. F. Valiati, and W. P. G. Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, Feb. 2013, doi: 10.1016/j.eswa.2012.07.059.

[24] 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.

[25] 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. (EMNLP)*, Sep. 2015, pp. 1422–1432, doi: 10.18653/v1/d15-1167.

[26] 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, p. e1253, Jul. 2018.

[27] 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, doi: 10.1016/j.eswa.2018.10.003.

[28] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2016, pp. 606–615, doi: 10.18653/v1/d16-1058.

[29] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. 26th Int. Conf. Comput. Linguistics (COLING)*, 2016, pp. 3298–3307.

[30] H. Zhao, Z. Lu, and P. Poupart, "Self-adaptive hierarchical sentence model," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jan. 2015, pp. 4069–4076.

[31] Z. Teng, D. T. Vo, and Y. Zhang, "Context-sensitive lexicon features for neural sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2016, pp. 1629–1638, doi: 10.18653/v1/d16-1169.

[32] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to generate reviews and discovering sentiment," 2017, *arXiv:1704.01444*. [Online]. Available: http://arxiv.org/abs/1704.01444

[33] A. B. Dieng, C. Wang, J. Gao, and J. Paisley, "TopicRNN: A recurrent neural network with long-range semantic dependency," 2016, *arXiv:1611.01702*. [Online]. Available: http://arxiv.org/abs/1611.01702

[34] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[35] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2016, pp. 214–224, doi: 10.18653/v1/d16-1021.

[36] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[38] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, vol. 1, 2014, pp. 655–665, doi: 10.3115/v1/p14-1062.

[39] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," 2015, *arXiv:1512.01100*. [Online]. Available: https://arxiv.org/abs/1512.01100

[40] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751, doi: 10.3115/v1/d14-1181.

[41] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process. (ACL-IJCNLP)*, vol. 1, 2015, pp. 1556–1566, doi: 10.3115/v1/p15-1150.

[42] 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.*, 2012, pp. 1201–1211.

[43] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 151–161.

[44] J. Zhao, "Adaptive learning of local semantic and global structure representations for text classification," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 2033–2043.

[45] A. Mishra, K. Dey, and P. Bhattacharyya, "Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 377–387, doi: 10.18653/v1/P17-1035.

[46] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, no. 1, 2011, pp. 513–520.

[47] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, vol. 4, 2014, pp. 2931–2939.

[48] D. T. Vo and Y. Zhang, "Target-dependent Twitter sentiment classification with rich automatic features," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jan. 2015, pp. 1347–1353.

[49] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2004, pp. 168–177, doi: 10.1145/1014052.1014073.

[50] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 440–447.

[51] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2014, pp. 49–54, doi: 10.3115/v1/p14-2009.

[52] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proc. 12th Int. Conf. World Wide Web (WWW)*, 2003, pp. 519–528, doi: 10.1145/775152.775226.

[53] Y. Xiao and K. Cho, "Efficient character-level document classification by combining convolution and recurrent layers," 2016, *arXiv:1602.00367*. [Online]. Available: http://arxiv.org/abs/1602.00367

[54] Q. Miao, Q. Li, and R. Dai, "AMAZING: A sentiment mining and retrieval system," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 7192–7198, Apr. 2009, doi: 10.1016/j.eswa.2008.09.035.

[55] M. Taboada, C. Anthony, and K. D. Voll, "Methods for creating semantic orientation dictionaries," in *Proc. LREC*, 2006, pp. 427–432.

[56] H. Kang, S. J. Yoo, and D. Han, "Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 6000–6010, Apr. 2012, doi: 10.1016/j.eswa.2011.11.107.

[57] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003, doi: 10.1162/153244303322533223.

[58] 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. (ICML)*, 2008, pp. 160–167.

[59] J. Yu and J. Jiang, "Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 236–246, doi: 10.18653/v1/d16-1023.

[60] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 496–509, Feb. 2016.

[61] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2011, pp. 142–150.

[62] X. Wang, Y. Liu, C. Sun, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long short-term memory," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 1343–1353, doi: 10.3115/v1/p15-1130.

[63] P. Liu, S. Joty, and H. Meng, "Fine-grained opinion mining with recurrent neural networks and word embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 1433–1443, doi: 10.18653/v1/d15-1168.

[64] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl.-Based Syst.*, vol. 108, pp. 42–49, Sep. 2016, doi: 10.1016/j.knosys.2016.06.009.

[65] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–11.

[66] H. Wu, Y. Gu, S. Sun, and X. Gu, "Aspect-based opinion summarization with convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3157–3163, doi: 10.1109/IJCNN.2016.7727602.

[67] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2014, pp. 1555–1565.

[68] H. Zhou, L. Chen, F. Shi, and D. Huang, "Learning bilingual sentiment word embeddings for cross-language sentiment classification," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 430–440.

[69] W. Zhang, Q. Yuan, J. Han, and J. Wang, "Collaborative multi-level embedding learning from reviews for rating prediction," in *Proc. IJCAI*, 2016, pp. 2986–2992.

[70] J. Vandewalle and D. Callaerts, "Singular value decomposition: A powerful concept and tool in signal processing," in *Mathematics in Signal Processing*. 1990, pp. 539–560.

[71] L. Wang and R. Xia, "Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 502–510.

[72] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 103–112, doi: 10.3115/v1/n15-1011.

[73] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1650–1659, doi: 10.18653/v1/d16-1171.

[74] Y. Zhang, Z. Zhang, D. Miao, and J. Wang, "Three-way enhanced convolutional neural networks for sentence-level sentiment classification," *Inf. Sci.*, vol. 477, pp. 55–64, Mar. 2019, doi: 10.1016/j.ins.2018.10.030.

[75] C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. 25th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2014, pp. 69–78.

[76] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2016, pp. 225–230, doi: 10.18653/v1/p16-2037.

[77] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 2428–2437.

[78] C. Guggilla, T. Miller, and I. Gurevych, "CNN- and LSTM-based claim classification in online user comments," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 2740–2751.

[79] Z. Guan, L. Chen, W. Zhao, Y. Zheng, S. Tan, and D. Cai, "Weakly-supervised deep learning for customer review sentiment classification," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jan. 2016, pp. 3719–3725.

[80] B. Huang and K. Carley, "Parameterized convolutional neural networks for aspect level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1091–1096, doi: 10.18653/v1/d18-1136.

[81] G. Letarte, F. Paradis, P. Giguère, and F. Laviolette, "Importance of self-attention for sentiment analysis," in *Proc. EMNLP Workshop BlackboxNLP, Analyzing Interpreting Neural Netw. NLP*, 2018, pp. 267–275, doi: 10.18653/v1/w18-5429.

[82] Z.-Y. Dou, "Capturing user and product information for document level sentiment analysis with deep memory network," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 521–526, doi: 10.18653/v1/d17-1054.

[83] J. Xu, D. Chen, X. Qiu, and X. Huang, "Cached long short-term memory neural networks for document-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1660–1669, doi: 10.18653/v1/d16-1172.

[84] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL HLT)*, 2016, pp. 1480–1489, doi: 10.18653/v1/n16-1174.

[85] X. Zhou, X. Wan, and J. Xiao, "Attention-based LSTM network for cross-lingual sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2016, pp. 247–256, doi: 10.18653/v1/d16-1024.

[86] Z. Li, Y. Zhang, Y. Wei, Y. Wu, and Q. Yang, "End-to-end adversarial memory network for cross-domain sentiment classification," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2237–2243, doi: 10.24963/ijcai.2017/311.

[87] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting LSTM networks for semi-supervised text classification via mixed objective function," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6940–6948, doi: 10.1609/aaai.v33i01.33016940.

[88] M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto, "Interpretable adversarial perturbation in input embedding space for text," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4323–4330, doi: 10.24963/ijcai.2018/601.

[89] R. Kiros, "Skip-thought vectors," in *Proc. Adv. Neural Inf. Process. Syst.*, no. 786, 2015, pp. 3294–3302.

[90] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.

[91] M. Huang, Q. Qian, and X. Zhu, "Encoding syntactic knowledge in neural networks for sentiment classification," *ACM Trans. Inf. Syst.*, vol. 35, no. 3, pp. 1–27, Jun. 2017, doi: 10.1145/3052770.

[92] Q. Qian, M. Huang, J. Lei, and X. Zhu, "Linguistically regularized LSTM for sentiment classification," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1679–1689, doi: 10.18653/v1/P17-1154.

[93] A. Nie, E. D. Bennett, and N. D. Goodman, "DisSent: Sentence representation learning from explicit discourse relations," 2017, *arXiv:1710.04334*. [Online]. Available: http://arxiv.org/abs/1710.04334

[94] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal, "Learning general purpose distributed sentence representations via large scale multi-task learning," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[95] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[96] C. Li, X. Guo, and Q. Mei, "Deep memory networks for attitude identification," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2017, pp. 671–680, doi: 10.1145/3018661.3018714.

[97] M. Zhang, Y. Zhang, and D. Vo, "Gated neural networks for targeted sentiment analysis," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2014, pp. 3087–3093.

[98] H. T. Nguyen and M. Le Nguyen, "Effective attention networks for aspect-level sentiment classification," in *Proc. 10th Int. Conf. Knowl. Syst. Eng. (KSE)*, Nov. 2018, pp. 25–30, doi: 10.1109/KSE.2018.8573324.

[99] Z. Zhang, L. Wang, Y. Zou, and C. Gan, "The optimally designed dynamic memory networks for targeted sentiment classification," *Neurocomputing*, vol. 309, pp. 36–45, Oct. 2018, doi: 10.1016/j.neucom.2018.04.068.

[100] S. Ruder, P. Ghaffari, and J. G. Breslin, "A hierarchical model of reviews for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 999–1005, doi: 10.18653/v1/d16-1103.

[101] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 5013–5014.

[102] J. Liu and Y. Zhang, "Attention modeling for targeted sentiment," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL)*, vol. 2, 2017, pp. 572–577, doi: 10.18653/v1/e17-2091.

[103] Y. Tay, L. A. Tuan, and S. C. Hui, "Dyadic memory networks for aspect-based sentiment analysis," in *Proc. Int. Conf. Inf. Knowl. Manag.*, Nov. 2017, pp. 107–116, doi: 10.1145/3132847.3132936.

[104] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2017, pp. 452–461, doi: 10.18653/v1/d17-1047.

[105] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé, III, "Deep unordered composition rivals syntactic methods for text classification," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 1681–1691.

[106] O. Irsoy and C. Cardie, "Deep recursive neural networks for compositionality in language," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, Jan. 2014, pp. 2096–2104.

[107] L. Dong, F. Wei, S. Liu, M. Zhou, and K. Xu, "A statistical parsing framework for sentiment classification," *Comput. Linguistics*, vol. 41, no. 2, pp. 293–336, Jun. 2015.

[108] Q. Qian, B. Tian, M. Huang, Y. Liu, X. Zhu, and X. Zhu, "Learning tag embeddings and tag-specific composition functions in recursive neural network," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process. (ACL-IJCNLP)*, vol. 1, 2015, pp. 1365–1374, doi: 10.3115/v1/p15-1132.

[109] L. Dong, F. Wei, M. Zhou, and K. Xu, "Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis," in *Proc. Nat. Conf. Artif. Intell.*, vol. 2, 2014, pp. 1537–1543.

[110] Z. Cheng, C. Yuan, J. Li, and H. Yang, "TreeNet: Learning sentence representations with unconstrained tree structure," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 4005–4011, doi: 10.24963/ijcai.2018/557.

[111] R. Johnson and T. Zhang, "Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level," 2016, pp. 1–7, *arXiv:1609.00718*. [Online]. Available: https://arxiv.org/abs/1609.00718

[112] J. Du, L. Gui, Y. He, R. Xu, and X. Wang, "Convolution-based neural attention with applications to sentiment classification," *IEEE Access*, vol. 7, pp. 27983–27992, 2019, doi: 10.1109/ACCESS.2019.2900335.

[113] M. Tutek and J. Šnajder, "Iterative recursive attention model for interpretable sequence classification," 2019, pp. 249–257, *arXiv:1808.10503*. [Online]. Available: https://arxiv.org/abs/1808.10503, doi: 10.18653/v1/w18-5427.

[114] J. Choi, K. M. Yoo, and S. Lee, "Learning to compose task-specific tree structures," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[115] W. Yin and H. Schütze, "Multichannel variable-size convolution for sentence classification," in *Proc. 19th Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2015, pp. 204–214, doi: 10.18653/v1/k15-1021.

[116] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 1014–1023, doi: 10.3115/v1/p15-1098.

[117] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing deep neural networks with logic rules," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 4, 2016, pp. 2410–2420, doi: 10.18653/v1/p16-1228.
[118] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[119] 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
[120] Y. Tay, L. A. Tuan, and S. C. Hui, "Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5956–5963.
[121] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, and H. Wang, "Aspect-level sentiment classification with HEAT (hierarchical attention) network," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 97–106.
[122] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. Nat. Conf. Artif. Intell.*, vol. 3, 2015, pp. 2267–2273.
[123] L. Shastri and C. Wendelken, "Learning structured representations," *Neurocomputing*, vols. 52–54, pp. 363–370, Jun. 2003, doi: 10.1016/S0925-2312(02)00840-8.
[124] B. Krause, I. Murray, S. Renals, and L. Lu, "Multiplicative LSTM for sequence modelling," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, vol. 2016, pp. 1–11.
[125] K. M. Hermann and P. Blunsom, "The role of syntax in vector space models of compositional semantics," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, vol. 1, 2013, pp. 894–904.
[126] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259
[127] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Rec. Syst. (RecSys)*, 2013, pp. 165–172.
[128] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 27–35, doi: 10.3115/v1/s14-2004.
[129] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 486–495, doi: 10.18653/v1/s15-2082.
[130] M. Pontiki, "SemEval-2016 task 5?: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 19–30.
[131] 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.
[132] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, Dec. 2016.
[133] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1081–1088.
[134] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL)*, 2017, pp. 427–431, doi: 10.18653/v1/e17-2068.
[135] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/d14-1162.
[136] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 384–394.
[137] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2012, pp. 873–882.
[138] G. Weikum, "Foundations of statistical natural language processing," *ACM SIGMOD Rec.*, vol. 31, no. 3, pp. 37–38, Sep. 2002, doi: 10.1145/601858.601867.
[139] M. Eirinaki, S. Pisal, and J. Singh, "Feature-based opinion mining and ranking," *J. Comput. Syst. Sci.*, vol. 78, no. 4, pp. 1175–1184, Jul. 2012, doi: 10.1016/j.jcss.2011.10.007.
[140] C. Yang, H. Zhang, B. Jiang, and K. Li, "Aspect-based sentiment analysis with alternating coattention networks," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 463–478, May 2019, doi: 10.1016/j.ipm.2018.12.004.
[141] J. Cohen, "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit.," *Psychol. Bull.*, vol. 70, no. 4, pp. 213–220, 1968, doi: 10.1037/h0026256.
[142] N. Novielli, D. Girardi, and F. Lanubile, "A benchmark study on sentiment analysis for software engineering research," in *Proc. 15th Int. Conf. Mining Softw. Repositories (MSR)*, 2018, pp. 364–375, doi: 10.1145/3196398.3196403.
[143] S. Kiyono, J. Suzuki, and K. Inui, "Mixture of expert/imitator networks: Scalable semi-supervised learning framework," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4073–4081, doi: 10.1609/aaai.v33i01.33014073.
[144] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguistics*, vol. 37, no. 1, pp. 9–27, Mar. 2011, doi: 10.1162/coli_a_00034.
[145] Z. Wang, W. Hamza, and L. Song, "*k*-nearest neighbor augmented neural networks for text classification," 2017, *arXiv:1708.07863*. [Online]. Available: [Online]. Available: https://arxiv.org/abs/1708.07863
[146] S. Zheng and R. Xia, "Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention," 2018, *arXiv:1802.00892*. [Online]. Available: http://arxiv.org/abs/1802.00892
[147] C. Qiao, B. Huang, G. Niu, D. Li, D. Dong, W. He, D. Yu, and H. Wu, "A new method of region embedding for text classification," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2012, pp. 1–12.
[148] S. Brahma, "Suffix bidirectional long short-term memory," 2018, *arXiv:1805.07340v1*. [Online]. Available: https://arxiv.org/abs/1805.07340v1
[149] Z. Zhang and M. Lan, "ECNU: Extracting effective features from multiple sequential sentences for target-dependent sentiment analysis in reviews," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 1–6, doi: 10.18653/v1/s15-2125.
[150] 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
[151] B. T. Do, "Aspect-based sentiment analysis using bitmask bidirectional long short term memory networks," in *Proc. 31st Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, 2018, pp. 259–264.
[152] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 562–570, doi: 10.18653/v1/P17-1052.
[153] M. Dragoni, M. Federici, and A. Rexha, "An unsupervised aspect extraction strategy for monitoring real-time reviews stream," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 1103–1118, May 2019, doi: 10.1016/j.ipm.2018.04.010.
[154] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using LSTM for region embeddings," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, vol. 2, 2016, pp. 794–802.
[155] J. Saias, "Sentiue: Target and aspect based sentiment analysis in SemEval-2015 task 12," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 1–5, doi: 10.18653/v1/s15-2130.
[156] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019, doi: 10.1016/j.neucom.2019.01.078.
[157] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*. [Online]. Available: http://arxiv.org/abs/1802.05365
[158] Y. Li, B. Wei, Y. Liu, L. Yao, H. Chen, J. Yu, and W. Zhu, "Incorporating knowledge into neural network for text representation," *Expert Syst. Appl.*, vol. 96, pp. 103–114, Apr. 2018, doi: 10.1016/j.eswa.2017.11.037.
[159] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 328–339, doi: 10.18653/v1/p18-1031.
[160] C. Brun, D. N. Popa, and C. Roux, "XRCE: Hybrid classification for aspect-based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 1–5, doi: 10.3115/v1/s14-2149.
[161] S. M. Michael, B. M. Chien, and D. M. Lubman, "Detection of electrospray ionization using a quadrupole ion trap storage/reflectron time-of-flight mass spectrometer," *Anal. Chem.*, vol. 65, no. 19, pp. 2614–2620, Oct. 1993, doi: 10.1021/ac00067a012.
[162] T. Munkhdalai and H. Yu, "Neural semantic encoders," in *Proc. Conf. Assoc. Comput. Linguistics Meeting*, vol. 1, 2017, p. 397.
[163] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2015, pp. 919–927.

[164] B. Li, Z. Zhao, T. Liu, P. Wang, and X. Du, "Weighted neural bag-of-n-grams model: New baselines for text classification," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 1591–1600.
[165] M. Looks, M. Herreshoff, D. L. Hutchins, and P. Norvig, "Deep learning with dynamic computation graphs," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–12.
[166] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3079–3087.
[167] T. Kim, J. Choi, D. Edmiston, S. Bae, and S. Lee, "Dynamic compositionality in recursive neural networks with structure-aware tag representations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6594–6601, doi: 10.1609/aaai.v33i01.33016594.
[168] Y. Tay, L. A. Tuan, and S. C. Hui, "Recurrently controlled recurrent networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2018, pp. 4731–4743.
[169] T. Lei, Z. Shi, D. Liu, L. Yang, and F. Zhu, "A novel CNN-based method for question classification in intelligent question answering," in *Proc. Int. Conf. Algorithms, Comput. Artif. Intell. (ACAI)*, 2018, pp. 1–6, doi: 10.1145/3302425.3302483.
[170] T. Munkhdalai and H. Yu, "Neural tree indexers for text understanding," in *Proc. Conf. Assoc. Comput. Linguistics Meeting*, vol. 1, 2017, p. 11.
[171] T. Zhang, M. Huang, and L. Zhao, "Learning structured representation for text classification via reinforcement learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 6053–6060.
[172] L. Zhining, G. Xiaozhuo, Z. Quan, and X. Taizhong, "Combining statistics-based and CNN-based information for sentence classification," in *Proc. IEEE 28th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2016, pp. 1012–1018, doi: 10.1109/ICTAI.2016.0156.
[173] T. Liu, S. Yu, B. Xu, and H. Yin, "Recurrent networks with attention and convolutional networks for sentence representation and classification," *Appl. Intell.*, vol. 48, no. 10, pp. 3797–3806, 2018, doi: 10.1007/s10489-018-1176-4.
[174] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 440–450, doi: 10.18653/v1/p18-1041.
[175] C. Xu, Y. Wu, and Z. Liu, "Multimodal fusion with global and local features for text classification," in *Proc. Int. Conf. Neural Inf. Process.*, in Lecture Notes in Computer Science, vol. 10634, 2017, pp. 124–134, doi: 10.1007/978-3-319-70087-8_14.
[176] Y. Zhang, D. Shen, G. Wang, Z. Gan, R. Henao, and L. Carin, "Deconvolutional paragraph representation learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 4170–4180.
[177] F. Ali, D. Kwak, P. Khan, S. El-Sappagh, A. Ali, S. Ullah, K. H. Kim, and K.-S. Kwak, "Transportation sentiment analysis using word embedding and ontology-based topic modeling," *Knowl.-Based Syst.*, vol. 174, pp. 27–42, Jun. 2019.
[178] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2015, pp. 959–962.
[179] G. Mesnil, T. Mikolov, M. Ranzato, and Y. Bengio, "Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews," 2014, *arXiv:1412.5335*. [Online]. Available: http://arxiv.org/abs/1412.5335
[180] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–19.
[181] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450
[182] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabelled data," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL HLT)*, 2016, pp. 1367–1377, doi: 10.18653/v1/n16-1162.

ALHASSAN MABROUK received the B.Sc. degree from Beni-Suef University, Egypt, in 2016. He is currently a Teaching Assistant with the Mathematics and Computer Science Department, Beni-Suef University. His research interests include sentiment analysis and machine learning.

REBECA P. DÍAZ REDONDO is currently an Associate Professor with the Telematics Engineering Department, University of Vigo. She is also working on defining appropriate architectures for distributed and collaborative data analysis, especially thought for IoT solutions, where computation must be on the edge of the network (fog computing). She has participated in more than 40 projects and 25 works of technological transfer through contracts with companies and/or public institutions. She is also involved in the scientific and technical activities of several National, and European research and educative projects.

MOHAMMED KAYED received the M.Sc. degree in computer science from Minia University, Minia, Egypt, in 2002, and the Ph.D. degree in computer science from Beni-Suef University, Beni-Suef, Egypt, in 2007.

From 2005 to 2006, he was a Research and Teaching Assistant with the Department of Computer Science and Information Engineering, National Central University, Taiwan. Since 2007, he has been an Assistant Professor with the Math&CS Department, Faculty of Science, Beni-Suef University. He is currently an Associate Professor and the Head of the Computer Science Department, Faculty of Computer and Artificial Intelligence, Beni-Suef University. He is the author of more than 25 articles. His research interests include web mining, opinion mining, information extraction, and information retrieval.