IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 10, 2019, accepted July 15, 2019, date of publication August 12, 2019, date of current version August 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934529

Semantic-Emotion Neural Network for Emotion Recognition From Text

ERDENEBILEG BATBAATAR[1], MEIJING LI[2], AND KEUN HO RYU[3,4], (Life Member, IEEE)
[1]School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, South Korea
[2]College of Information Engineering, Shanghai Maritime University, Shanghai 200136, China
[3]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
[4]Database and Bioinformatics Laboratory, School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, South Korea

Corresponding author: Keun Ho Ryu (khryu@tdtu.edu.vn)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, Information and Communication Technology (ICT) and Future Planning under Grant 2017R1A2B4010826 and Grant NRF-2019K2A9A2A06020672, and in part by the National Natural Science Foundation of China in People's Republic of China under Grant 61702324.

ABSTRACT Emotion detection and recognition from text is a recent essential research area in Natural Language Processing (NLP) which may reveal some valuable input to a variety of purposes. Nowadays, writings take many forms of social media posts, micro-blogs, news articles, customer review, etc., and the content of these short-texts can be a useful resource for text mining to discover an unhide various aspects, including emotions. The previously presented models mainly adopted word embedding vectors that represent rich semantic/syntactic information and those models cannot capture the emotional relationship between words. Recently, some emotional word embeddings are proposed but it requires semantic and syntactic information vice versa. To address this issue, we proposed a novel neural network architecture, called SENN (Semantic-Emotion Neural Network) which can utilize both semantic/syntactic and emotional information by adopting pre-trained word representations. SENN model has mainly two sub-networks, the first sub-network uses bidirectional Long-Short Term Memory (BiLSTM) to capture contextual information and focuses on semantic relationship, the second sub-network uses the convolutional neural network (CNN) to extract emotional features and focuses on the emotional relationship between words from the text. We conducted a comprehensive performance evaluation for the proposed model using standard real-world datasets. We adopted the notion of Ekman's six basic emotions. The experimental results show that the proposed model achieves a significantly superior quality of emotion recognition with various state-of-the-art approaches and further can be improved by other emotional word embeddings.

INDEX TERMS Emotion recognition, natural language processing, deep learning.

I. INTRODUCTION

Emotion recognition will play a promising role in the field of artificial intelligence and human-computer interaction [1]. Various types of techniques are used to detect emotions from a human being like facial expressions [2], body movements [3], blood pressure [4], heartbeat [5] and textual information [6]. In computational linguistics, the detection of human emotions in a text is becoming increasingly important from an applicative point of view. Nowadays within the internet, there's an enormous amount of textual data. It's fascinating to extract emotion from various goals like those of business [7]. As an example, in luxury merchandise, the emotional aspect as brand, individuality and prestige for purchasing confirmations, are a lot necessary than other aspects such as technical, functional or price [8]. There are basic emotion theories that have been developed on how some emotions are considered more than others [9], [10]. This study adopted basic six emotions [9] including joy, fear, anger, sadness, surprise and disgust.

There are many works which have achieved reasonable results in the field of emotion recognition from text. Improving the previous result and emotion recognition using real-world data still remains a huge challenge for several reasons. Most machine learning methods overly rely on handcrafted features which require lots of manual design and adjustment,

The associate editor coordinating the review of this article and approving it for publication was Irene Amerini.

and it is time-consuming and cost intensive. Though the problem is helped greatly by the proposal of deep learning in recent years.

Word embeddings are widely used for many NLP tasks, such as sentiment analysis [11], question answering [12], and machine translation [13]. Existing word embedding learning approaches mostly represent each word by predicting the target word through its context [14], [15] and map words of similar semantic roles into nearby points in the embedding space. For example, the words ''good'' and ''bad'' are semantically similar and mapped into embedding space closely. It is confused, however, in emotional condition. Recently, some emotion embeddings are proposed for solving this issue and achieved better performance in emotion-related tasks [16]–[19].

Previous studies mostly used semantic based word embeddings and achieved good results [20]–[22] by training a single model which can adopt either semantic or emotion word embeddings. As mentioned, these neural network approaches cannot encode and learn both semantic and emotional relationship in short text efficiently.

In order to address the above limitations, this paper proposed a novel neural network architecture, called semantic-emotion neural network (SENN) which can utilize both semantic and emotion information by adopting existing pre-trained word embeddings. We divided SENN into two sub-networks. The first network uses BiLSTM to capture semantic information map it into semantic-sentence space, the second network uses CNN to capture emotion information and map it into emotion-sentence space. CNN is supposed to be good at extracting position invariant features such as emotion terms and BiLSTM at modeling units in sequence of long semantics in whole sentence. Then we combine the final representation together for further emotion recognition. Experimental results show that the SENN model outperforms most of the baseline methods and state-of-the-art approaches. The main contributions of this work are as follows:

a) Semantic and emotion word embeddings are adopted separately in two sub-networks for same text input.
b) Under the framework of deep neural network, we use BiLSTM and CNN for designing semantic and emotion sentence encoder respectively. BiLSTM is designed to capture contextual information and CNN is designed to extract emotional information effectively.
c) A novel dual neural network model is proposed. We respectively use BiLSTM and CNN for encoding semantic and emotion text. Then we combine the final representation by concatenating semantic and emotion sentence encoding for further emotion recognition.
d) To get a better knowledge of semantic and emotion information on a specific dataset, we used the fine-tuning technique on pre-trained word embeddings which improves the performance of emotion recognition models from text efficiently. Then we concatenated the sentence-level encoded vectors to recognize emotion from the text.

The rest of the paper is organized as follows. Section II presents related works. Section III describes the detail of the proposed SENN model architecture. Section IV outlines the experimental setup, and Section V discusses the empirical results and analysis. Finally, Section VI presents the conclusion and introduces the next research direction.

II. RELATED WORK

This section reviews recent advances in emotion recognition analysis. We roughly categorize the existing studies into two types: 1) Emotion recognition based on the traditional method; 2) Emotion recognition based on deep learning approaches.

A. EMOTION RECOGNITION BASED ON TRADITIONAL METHOD

The traditional methods on emotion recognition analysis can be roughly subdivided into two categories: 1) lexicon-based method; 2) machine learning method. Lexicon based methods utilize pre-defined lists of terms that are categorized according to different emotions [23]. On the one hand, these lexicons are often compiled manually, a fact which can later be exploited for keyword matching. For instance, the NRC Word-Emotion Association lexicon was derived analogously but with the help of crowdsourcing rather than involving experts from the field of psychology research, they annotated around 14000 words in the English language [24]. Another popular emotion lexicon is WordNet-Affect dictionary [25]. They tried to create a lexical representation of effective knowledge by starting from the WordNet lexical database [26]. WordNet-Affect dictionary starts with a set of seed words labelled as effect and then assigns scores to all other words based on their proximity to the seed words. Another attempt to generate an emotional lexicon is DepecheMood [27]. They used crowdsourcing to annotate 35,000 words. They showed that lexicons could be used in several approaches in sentiment analysis, as features for classification in machine learning methods [28], or to generate an affect score for each sentence, based on the scores of the words which are higher in the parse tree [29]. Other emotional lexicons frequently used in the literature are LIWC lexicon [30] consisting of 6,400 words annotated for emotions, and also ANEW (Affective Norm for English Words) [31]. This dataset has near 2,000 words which have been annotated based on the dimensional model of emotions, with three dimensions of valance, arousal and dominance. Lexicon based approaches are generally known for their straightforward use and out-of-the-box functionality. However, manual labelling is error-prone, costly, and inflexible as it impedes domain customization. Conversely, the vocabulary from the heuristics is limited to a narrow set of dimensions that were selected a priori and, as a result, this procedure has difficulties when generalizing to other emotions [32].

Machine learning can infer decision rules for recognizing emotions based on a corpus of training samples with explicit labels [33], [34]. Previous researches have experimented with different models for inferring emotion from narrative materials. Examples include methods that explicitly exploit the flexibility of machine learning such as Naïve Bayes (NB) [35], Random Forest (RF) [36], [37], Support Vector Machine (SVM) [33], [35], [38]–[40], and Logistic Regression (LR) [37]. All of which have commonly been deployed in literature. These classifiers are occasionally, but infrequently, restricted to the subset of affect cues from emotion lexicons [41]. The more common approaches rely upon general linguistic features, bag-of-words (BOW) with subsequent term frequency-inverse document frequency (TF-IDF) weighting [42], [43]. However, these features are often not suitable for document distances due to their frequent near-orthogonality [44], [45].

B. EMOTION RECOGNITION BASED ON DEEP LEARNING

In the following, we discuss the few attempts at applying deep learning to emotion recognition, but find that actual performance evaluations are scarce. CNN with a sliding window and subsequent max-pooling are used to predict aggression expressed through NLP [36]. However, this approach is subject to several limitations as the network is designed to handle only a single dimension and it is thus unclear how it generalizes across multi-class predictions or even regression tasks that appear in dimensional emotion models. Even though the approach utilizes a deep network, its network architecture can only handle texts of a predefined size, analogous to traditional machine learning. In this respect, it differs from recurrent networks, which iterate over sequences and thus can handle texts of arbitrary size.

There are many recurrent neural network (RNN) methods that are introduced for emotion recognition tasks. Due to the lack of emotion-labelled datasets, many supervised classification algorithms for emotions have been done on data gathered from microblog such as Twitter, using hashtags or emoticons as the emotion label for the data. The current state-of-the-art methods are introduced by using Gated Recurrent Unit (GRU) network [20] for fine-grained emotion recognition. They firstly built a large dataset for emotion recognition automatically from Twitter, then extended the classification to eight primary emotion dimensions situated in the psychological theory of emotion.

A Long-Short Term Memory (LSTM) [19] is utilized that is pre-trained with tweets based on the appearance of emoticons. However, this work does not report a comparison of their LSTM against a baseline from traditional machine learning. A different approach [46] utilizes a custom LSTM architecture in order to assign emotion labels to complete conversations in social media. However, this approach is tailored to the specific characteristics and emotions of this type of conversational-style data. In addition, the conclusion from their numerical experiments cannot be generalized to emotion recognition, since the authors labelled their dataset through a heuristic procedure and then reconstructed this heuristic with their classifier.

A BiLSTM is also utilized to recognize emotions in cross-lingual texts [47] which employs the cross-lingual feature and the lexical level feature to analyze texts with multi-lingual forms. To incorporate a context-dependent word, an attention-based BiLSTM model [48] is introduced, which helps to decide the importance of each word for the emotion recognition task. They used the three modalities such as text, emoji and images to encode different information to express emotions.

Deep learning based methods mostly uses distributed word vectors, commonly used methods are Word2Vec [14], GloVe [15], and FastText [49]. Word2Vec is one of the very first models to learn word representations from trillions of words with relatively low computational costs. It has significantly outperformed various n-gram models [50]. Later GloVe was released, which stands for global vectors for word representations. It was an improvement over Word2Vec as it trains on global co-occurrence counts instead of separate local context windows in Word2Vec. Lately, the FastText was created for classification and learning of word representation. Word2Vec and Glove treat words as the smallest atomic units. FastText uses a different approach where it treats each individual word as being made of n-gram characters and it is more powerful than other two models as it can effectively handle rare words which are not present in the dictionary. Moreover, contextualized word embeddings are proposed, called ELMo [51] and BERT [52], to incorporate context information and solve the polysemy issues in conventional word embeddings. However, these word embeddings are generalized on various tasks and limited to provide emotion information, therefore learning task-specific emotion embedding with the neural network has been proven to be effective. Emotion-enriched word embedding (EWE) is learned [17] on product reviews, with the much smaller corpus. This embedding could be easily applied to emotion-related tasks, which could largely overcome the limitations of emotion dictionary. It is, however, limited to provide a semantic and syntactic relationship.

Recently, other combined architectures are proposed for text-based emotion recognition tasks. For instance, the combined recurrent convolutional neural network (RCNN) approach [53] are introduced and achieved competitive results with fine-tuned contextual and emotional word embeddings [19], [51], [52]. The experimental results show that the fine-tuned GloVe embeddings perform noticeably better than contextual word embeddings, due to the emotion recognition task highly depends on emotion extraction and size of word dictionary.

There is some comparative study, which compared the traditional machine learning algorithms and deep learning based algorithms on large Twitter data [21]. They compared SVM, NB, LR, and RF algorithms with basic deep learning algorithms CNN and RNN with GloVe word embeddings. Generally, deep learning approaches outperformed the machine learning approaches in the emotion recognition task.

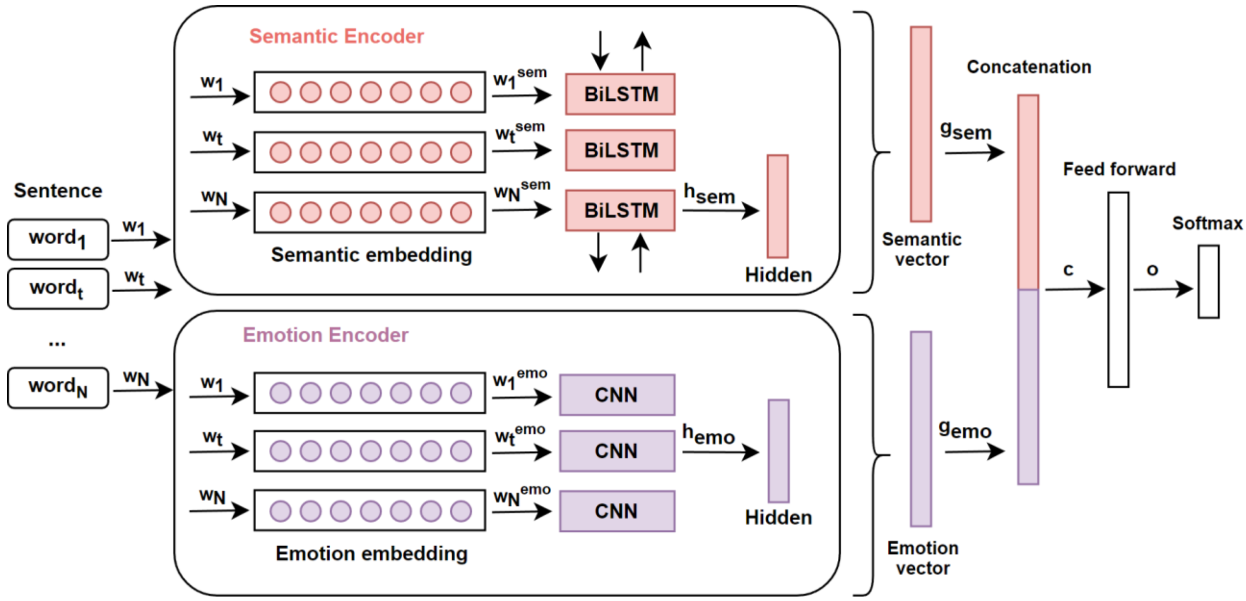

FIGURE 1. SENN model architecture.

However, they collected a large number of training data, twitter data is more biased and noisy compared than hand- aggregated corpus [54], which is released for emotion analysis research, the combined datasets are collected from different domains and different label set.

In this paper, we propose a novel SENN model with BiLSTM and CNN sub-networks to conduct the emotion recognition task. We used the word embeddings Word2Vec, GloVe, and FastText to capture the semantic relationship between words and emotional word embedding to extract emotional features and evaluated the SENN model and the other compared models.

III. SENN MODEL

In the paper, we propose a novel SENN model for emotion recognition from text and the structure is shown in Figure 1. It consists of two sub-networks: 1) BiLSTM network for semantic encoder between words and 2) CNN network for emotion encoder. The outputs of the sub-networks are used to recognize emotions from the text. Both of the two sub-networks are fed by the same sequence of N words and each word is transformed to a d dimensional word vector. Ultimately, the word embedding layer encodes the sequence representation as two matrices Z_{emo} (emotion word embedding) and Z_{sem} (semantic word embedding),

$$Z_{emo} = [w_1^{emo}, \ldots, w_t^{emo}, \ldots, w_N^{emo}] \in R^{N\times d} \quad (1)$$
$$Z_{sem} = [w_1^{sem}, \ldots, w_t^{sem}, \ldots, w_N^{sem}] \in R^{N\times d} \quad (2)$$

where w_t^e and w_t^s are the emotion and semantic word vectors of the word w_t in the sequence, respectively,

$$w_t^{emo} = [e_{t1}, \ldots, e_{tk}, \ldots, e_{td}] \quad (3)$$
$$w_t^{sem} = [s_{t1}, \ldots, s_{tk}, \ldots, s_{td}] \quad (4)$$

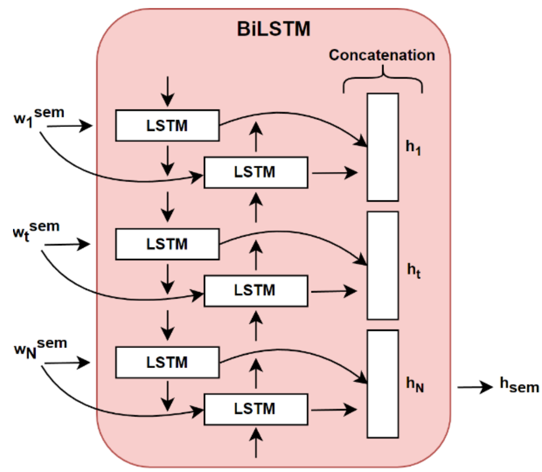

FIGURE 2. BiLSTM structure.

Since we have enough data and our task slightly differs from unsupervised task used for training pre-trained word embedding, we speculated that further fine-tuning during the training process might improve the embedding. Whether to do this or not was an additional parameter we optimized.

A. BILSTM NETWORK FOR SEMANTIC ENCODER

To better model the semantic information of text, we used bidirectional LSTM [55] to derive the hidden state of each word by summarizing the information from both forward and backward directions. An architecture of BiLSTM used in this paper is shown in Figure 2. An input for an LSTM is represented as Z_{sem} matrix and the LSTM is fed by semantic word vectors w_t^{sem}. Forward LSTM and backward LSTM are denoted as $\overrightarrow{LSTM}$ and $\overleftarrow{LSTM}$, whereas $\overrightarrow{LSTM}$ reads words from left to right and $\overleftarrow{LSTM}$ in reverse direction,

$$\overrightarrow{h_t} = \overrightarrow{LSTM}\left(w_t^{sem}, \overrightarrow{h_{t-1}}\right), \quad t \in [1, N] \tag{5}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}\left(w_t^{sem}, \overleftarrow{h_{t+1}}\right), \quad t \in [1, N] \tag{6}$$

We get a representation of each w_t^{sem} by concatenating the forward hidden state $\overrightarrow{h_t}$ and the backward hidden state $\overleftarrow{h_{t+1}}$.

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{7}$$

Finally, semantic sequence vector h_{sem} encoded from the last hidden state is fed into the hidden layer.

$$h_{sem} = h_N \tag{8}$$

$$g_{sem} = f\,(w_{sem}h_{sem} + b_{sem}) \tag{9}$$

where g_{sem} is the output of the semantic encoder, w_{sem} and b_{sem} are parameters of the f activation function.

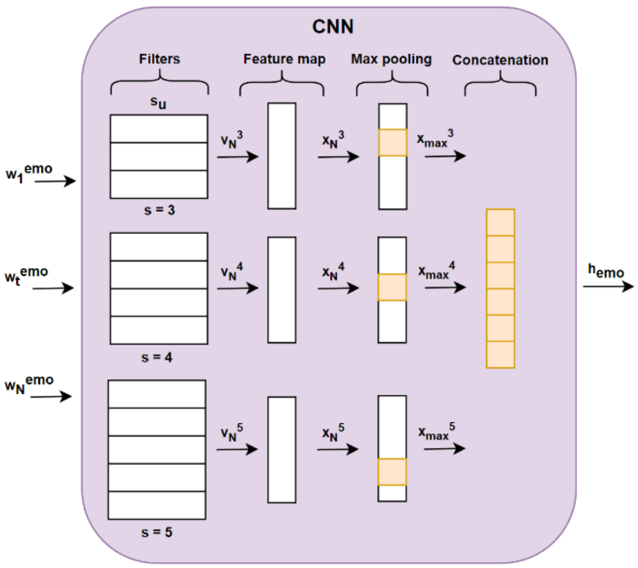

FIGURE 3. **CNN structure.**

B. CNN NETWORK FOR EMOTION ENCODER

To better extract emotion features from emotion-based word embeddings, we used CNN [56] to utilize layers with convolving filters that are applied to local features. An architecture of the CNN used in this paper is shown in Figure 3. An input for a CNN is represented as Z_{emo} matrix and the CNN is fed by emotion word vectors w_t^{emo}. Then the word embedding vectors are concatenated as the feature vector v of the sequence.

$$v = w_1^{emo} \oplus \ldots w_t^{emo} \ldots \oplus w_N{}^{emo} \tag{10}$$

where $\oplus$ is the concatenation operator of vectors. In the first convolution layer, convolution calculation is performed using employ multiple filters with variable window size s and generate a local emotion feature vector x_i for each possible word window size. And the bias term $b \in R$ and transition matrix $W \in R^{s_u \times sN}$ are generated for each filter, where s_u is the number of hidden units in the convolution layer. Each convolution operation generates a new context local feature vector x_i^s in a word window s.

$$x_i^s = f(W \cdot v_{i:i+s-1} + b) \tag{11}$$

where f is a non-linear activation function and $v_{i:i+s-1}$ is the local vector from position i to position $i + s - 1$ in the vector v. The convolution filter generates a local feature mapping vector for each possible word window in the input sequence, which is followed by the completion of the convolution operation to generate a new vector that can be expressed as:

$$x^s = [x_1^s, \ldots, x_i^s, \ldots, x_{N-s+1}^s] \tag{12}$$

Afterwards the convolution operation, max pooling operation is employed on the new feature vector x_i^s generated by the convolution layer. Max pooling mapped the vector x_i^s to a fixed length vector. The length of the vector is a hyperparameter to be determined by the user and corresponds to the number of hidden units in the convolution layer. The local sentence features are integrated into all the features. For emotion recognition, the most decisive word or phrase is often only a few, but not uniformly scattered throughout the text. The max pooling is just some of the most discriminative language fragments. The max pooling selects the top number of features corresponding to multiple hidden layers so that the most important emotion feature information can be retained. At the same time, the sequence of words and the context information of each word are also taken into consideration in the pooling operation.

$$x_{max}^s = \max\{x_1^s, \ldots, x_i^s, \ldots, x_{N-s+1}^s\} \tag{13}$$

Since there are multiple feature maps, we have a vector after the pooling operation. All vectors which are output from the max-pooling layer are concatenated into a single feature vector h_{emo}.

$$h_{emo} = \left[x_{max}^s\right], \quad s \in [s_{min}, s_{max}] \tag{14}$$

Finally, the emotion sequence vector h_{emo} is fed into the hidden layer.

$$g_{emo} = f\,(w_{emo}h_{emo} + b_{emo}) \tag{15}$$

where g_{emo} is the output of the emotion encoder, w_{emo} and b_{emo} are parameters of the non-linear f activation function.

C. EMOTION RECOGNITION

Finally, after producing emotion encoding g_{emo} and semantic encoding g_{sem}, we concatenated the vectors as c and fed it into the feedforward layer,

$$c = [g_{emo}, g_{sem}] \tag{16}$$

$$o = f\,(w_o c + b_o) \tag{17}$$

where f is the feed-forward layer, w_o and b_o are weight and bias parameters respectively. o is the output of the feed-forward layer.

Softmax classifier takes the output at the last step and o serves as its input. As noted above, given sequence with N words, we predict the emotion y for each sequence. Emotion annotations of sequences are represented by $Y(Y = Y_1, Y_2, \ldots, Y_M)$. The predicted values y' can be calculated by:

$$p(y|X) = softmax(w_p o + b_p) \tag{18}$$

and

$$y' = \underset{y}{arg\ max}\, p(y|X) \tag{19}$$

where p is the predicted probability of emotion label, w_p and b_p are parameters of the classification layer. We then use the cross-entropy to train the loss function. We first derive the loss of each labelled sequence and the final loss is averaged over all the labelled sequences by the following equation:

$$Loss = -\frac{1}{M}\sum\nolimits_{m=1}^{M} Y_m \cdot \log p(y_n|X_n) \tag{20}$$

where the subscript n indicates the n^{th} input sequence.

IV. EXPERIMENTAL SETUP

A. EXPERIMENTAL ENVIRONMENT

In this paper, the experimental hardware platform is Intel Xeon E3, 32G memory, GTX 1080 Ti. The experimental software platform is Ubuntu 17.10 operating system and development environment is Python 3.5 programming language. The Pytorch library and the Scikit-learn library of python are used to build the proposed emotion recognition model and comparative experiments, respectively.

B. EVALUATION MEASURES

For evaluating the SENN model, an exact matching criterion was used to examine three different result types. False negative (FN) and False positives (FP) are incorrect negative and positive predictions. True positives (TP) results corresponded to correct positive predictions, which are actual correct predictions. The evaluation is based on the performance measures precision (P), recall (R) and F-score (F1). Recall denotes the percentage of correctly labelled positive results overall positive cases and is calculated as:

$$R = \frac{TP}{TP + FN} \tag{21}$$

$$P = \frac{TP}{TP + FP} \tag{22}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{23}$$

C. BASELINE

We then compare the proposed method with other baseline models in terms of the F1-score. For this purpose, we implement the following baseline models:

- **Naïve Bayes (NB)** [35]: A basic multinomial Naïve Bayes classifier based on probability theory.
- **Random Forest (RF)** [36]: A basic random forest classifier based on ensemble learning method. We built forests with 10, 100 and 500 trees.
- **Support Vector Machine (SVM)** [33]: A basic support vector classifier based on hyperplane separator. We tested the parameters: Penalty (1, 10 and 100), kernel (linear and RBF) and gamma (0.001 and 0.0001).
- **Logistic Regression (LR)** [37]: A basic logistic regression classifier based on statistic method. We tested the parameters: Regularizer (0.001, 0.01, 0.1, 1, 10, 100, 100) and penalty (l1 and l2).
- **Convolutional Neural Network (CNN)** [36]: Only convolutional neural networks are used. CNN models show the advantages of extracting complicated emotion features.
- **Long-Short Term Memory (LSTM)** [19]: A model used GRU or BiGRU. It utilizes the last hidden state for emotion recognition. The models show the advantages of learning contextual semantic knowledge.
- **Gated Recurrent Unit (GRU)** [20]: A model used GRU or BiGRU. It utilizes the last hidden state for emotion recognition. The models show the advantages of learning contextual semantic knowledge.
- **RCNN** [17]: A combination of CNN and LSTM. The LSTM layer is used before CNN layer.
- **CNN + LSTM** [57]: Another combination of CNN and LSTM. The CNN layer used before LSTM layer.

For machine learning methods, regarding representing texts, we used Bag-of-Words and Term Frequency-Inverse Document Frequency as the feature of each text. We used a grid search algorithm to find optimal parameters. And for deep learning based methods, regarding to representing texts, we used Word2Vec (3 billion 300 dimension word vectors), GloVe (840 billion 300 dimension word vectors), and FastText (2 million 300 dimension word vectors) semantic word embeddings and EWE (10 thousand 300 dimension word vectors) emotion word embedding. For a fair comparison, we used the same hyperparameters settings [20] in default shown in Table 1.

TABLE 1. Hyperparameter setting.

Parameter	CNN	BiLSTM
Learning rate	0.001	0.001
Batch size	128	128
Hidden dimension	256	-
Number of layers	2	-
Number of filters	-	100
Filter sizes	-	[3, 4, 5]
Early stopping patience	20	20
Dropout	0.5	0.5

D. EMOTION RECOGNITION DATASET

In the experiment, we used the emotion-annotated datasets [54] which are from multiple domains (dialogues, tweets, fairy tales, blogs, and news headlines). We choose ten emotion recognition datasets to create our experimental data.

TABLE 2. Datasets descriptions (D: DailyDialogs; C: CrowdFlower; T: TEC; TE: Tales-Emotions; I: ISEAR; E: EmoInt; ET: Electoral-Tweets; G: Grounded-Emotions; EC: Emotion-Cause; S: SSEC).

Emotion Label	D	C	T	TE	I	E	ET	G	EC	S
Joy	10,291	8,412	7,714	1,777	1,064	1,548	344	1,474	479	232
Fear	154	7,799	2,466	689	1,051	2,149	90	-	413	15
Sadness	1,053	4,697	3,619	902	1,012	1,458	31	1,010	565	20
Surprise	1,423	1,978	3,673	772	-	-	122	-	206	35
Anger	924	1,311	1,492	708	1,058	1,621	565	-	468	23
Disgust	315	164	714	368	1,056	-	1,630	-	95	10
Total	14,160	24,361	19,678	5,216	5,241	6,776	2,782	2,484	2,226	335

- **Dailydialogs (D):** The dataset which is built on conversations. The annotation schema follows Ekman and non-emotional sentences [58].
- **CrowdFlower (C)**: The twitter dataset published by CrowdFlower. The set of labels is non-standard (see details in [54]). The tweets are annotated via crowdsourcing [59].
- **TEC (T):** The twitter dataset which is built on social media. The annotation schema corresponds to Ekman's model of basic emotions [60].
- **Tales-Emotions (TE):** The tales dataset which is built on literature. The annotation schema consists of Ekman's six basic emotions [42].
- **ISEAR (I):** The dataset which is built on collecting questionnaires answered by people with different cultural backgrounds. The labels are joy, fear, anger, sadness, disgust, shame, and guilt [61].
- **EmoInt (E):** The dataset which is built on social media content. The tweets are annotated via crowdsourcing with intensities of anger, joy, sadness, and fear [62].
- **Electoral-Tweets (ET):** The twitter dataset which targets the domain of elections. The set of labels is non-standard (see details in [54]). The tweets are annotated via crowdsourcing [63].
- **Grounded-Emotions (G):** The dataset which is built on social media. The set of labels is happy and sad. The tweets are annotated by the authors [54].
- **Emotion-Cause (EC):** The dataset which is annotated both with emotions and their causes. The set of labels used for annotation consists of Ekman's basic emotions to which shame is added [64].
- **SSEC (S):** The stance sentiment emotion dataset which is an annotation of the "SemEval 2016 Twitter" stance and sentiment dataset. It is annotated via expert annotation with multiple emotion labels per tweet following Plutchik's fundamental emotions [65].

We only selected single label annotated sentences and removed the sentences with emotional intensity. We only selected Ekman's six emotions including joy, fear, sadness, surprise, anger, and disgust. Table 2 shows the dataset information in detail.

E. DATA PREPROCESSING

All user-generated data need to be preprocessed before classification. Since text are written by the general public, there is a possibility, indeed a reasonable probability, that a large number of casual words, abbreviations and short forms, special characters and spelling mistakes, are present in user-generated data. These add to the noise in the input data, which will be used for the classification in learning-based approaches. For reasons previously mentioned, it is important to clean user-generated data before classification. In this work, we clean the data without any subtasks like spelling mistakes, handling casual words, abbreviations and short forms. During preprocessing the following simple steps are followed for better performance of emotion recognition.

- All numbers and special character are removed.
- All twitter IDs (starts with @) are removed.
- All uppercase characters are changed into lowercase characters.

F. HYPERPARAMETER AND TRAINING

The hyperparameters in the proposed emotion recognition method include hidden layer size, number of layers, batch size, learning rate, dropout in BiLSTM and CNN. The hyperparameters with the best classification effect of the model are studied. We used Adam optimizer to update parameters while training. And we used dropout and an early stopping strategy with patience 20 to avoid overfitting and early stopping monitored weighted F1-score on validation sets. The experimental dataset is randomly divided into 90:10 training set and testing set. We used 10% of the training set as a validation set.

V. EXPERIMENTAL RESULT

In this section, we evaluate our approach and report empirical results. We compare the proposed SENN model with several baselines including traditional machine learning and deep learning methods on several emotion recognition datasets. The experimental results for emotion recognition are listed in Table 3. When comparing the performance of the three variants of SENN model, it is observed that FastText performs better than Word2Vec and GloVe word embeddings. One possible reason is that the FastText word embedding can capture the meaning of shorter words and allows the embeddings to understand suffixes and prefixes. It works well with rare words and out-of-vocabulary words.

Compared with traditional machine learning models, we proved that deep learning based models outperformed the machine learning models as reported in previous studies.

TABLE 3. Comparison with the baseline models. The best results are indicated in bold with (*) and the second best result are indicated in only bold.

Model		D	C	T	TE	I	E	ET	G	EC	S
NB	BOW	73.2	44.7	49.0	48.2	68.7	83.7	49.9	55.7	71.0	58.4
	TFIDF	61.7	40.0	34.9	24.6	67.8	74.5	44.3	49.7	54.1	58.4
RF	BOW	77.8	43.8	48.1	40.1	63.7	86.9	47.6	54.6	94.6	58.4
	TFIDF	77.0	43.8	46.5	38.2	64.9	86.4	51.0	52.6	92.0	58.0
SVM	BOW	80.1	47.5	53.5	52.6	63.2	86.2	52.4	48.8	95.1	58.4
	TFIDF	78.5	46.1	55.0	54.1	65.9	88.1	52.3	50.6	95.5	57.0
LR	BOW	79.8	46.1	53.8	49.0	69.9	88.2	52.8	53.0	96.9	54.3
	TFIDF	80.0	46.8	55.6	49.8	69.5	89.2	51.6	50.0	**98.7**	53.3
CNN	Word2Vec	81.8	50.6	59.1	57.7	73.1	89.0	51.2	50.7	97.8	54.3
	GloVe	81.9	49.1	57.6	57.3	73.5	89.1	52.1	57.7	96.3	53.9
	FastText	82.6	50.2	58.5	**62.3***	73.9	89.7	52.5	48.3	95.9	55.5
	EWE	81.7	50.6	59.3	58.4	73.6	86.4	48.4	56.0	95.9	56.5
GRU	Word2Vec	80.5	45.7	54.2	46.0	67.8	87.9	53.7	52.9	96.4	58.7
	GloVe	80.5	46.1	56.0	46.3	25.5	87.8	54.7	48.9	96.8	57.1
	FastText	81.9	47.5	53.5	33.3	70.3	88.0	51.8	54.2	95.9	63.5
	EWE	81.5	46.3	54.2	40.5	71.5	86.4	53.1	55.5	97.3	57.6
BiGRU	Word2Vec	81.0	48.4	55.8	56.8	71.6	89.3	49.8	50.6	95.9	57.7
	GloVe	83.3	48.6	57.6	53.9	71.8	89.1	47.7	54.4	96.4	61.8
	FastText	82.8	48.0	56.6	56.1	72.3	89.2	51.7	56.1	96.9	57.0
	EWE	83.2	49.8	55.8	57.0	71.1	89.7	46.8	54.3	95.9	**67.4**
LSTM	Word2Vec	78.5	45.8	51.9	43.2	23.7	87.0	50.1	50.4	93.7	58.7
	GloVe	79.8	45.3	54.3	34.2	69.3	88.2	51.7	53.0	96.9	58.8
	FastText	63.5	45.7	54.4	50.5	25.4	88.4	51.6	51.1	92.0	58.4
	EWE	77.0	42.9	55.3	35.3	21.8	88.8	51.1	51.7	94.0	55.9
BiLSTM	Word2Vec	80.2	49.3	54.6	53.1	71.0	89.7	51.0	53.5	93.1	55.3
	GloVe	82.9	48.6	57.4	53.5	72.2	89.8	51.3	51.6	97.3	65.0
	FastText	82.2	48.5	56.2	55.8	70.2	89.2	49.0	51.1	95.4	57.0
	EWE	83.7	48.4	58.4	55.8	72.0	88.1	51.6	52.9	95.5	49.2
RCNN	Word2Vec	80.8	50.3	57.0	53.2	72.1	89.3	52.5	51.6	98.2	61.8
	GloVe	83.2	49.5	56.9	61.0	70.8	88.5	53.3	52.8	**98.7**	60.3
	FastText	81.8	49.6	57.4	59.2	73.3	89.6	52.4	51.8	97.8	55.9
	EWE	83.6	48.0	57.3	56.2	73.1	88.1	**55.4**	50.5	95.5	58.0
CNN LSTM	Word2Vec	82.1	49.7	56.4	56.9	71.6	87.6	47.6	49.6	96.4	55.4
	GloVe	83.3	49.8	56.5	55.4	72.4	88.6	49.4	56.0	97.3	55.3
	FastText	82.2	**51.0**	57.9	54.9	72.2	89.9	48.5	48.7	95.9	46.7
	EWE	82.7	50.0	57.3	54.9	73.5	89.0	49.3	52.3	95.9	48.1
SENN	Word2Vec + EWE	84.2	50.8	**59.7**	60.3	73.7	**91.0***	**55.4**	**58.4**	96.9	65.0
	GloVe + EWE	**84.3**	50.5	**59.7**	60.9	**74.6***	**90.4**	54.6	**59.3***	**98.8***	64.2
	FastText + EWE	**84.8***	**51.1***	**61.3***	**61.7**	**74.5**	**90.4**	**56.3***	55.6	98.3	**70.8***

Logistic regression and support vector machine shows the comparative result using bag-of-word and tf-idf vectors.

Compared with the state-of-the-art models in emotion classification, SENN gives the best performance on nine out of ten datasets except Tales-Emotion dataset. It performs F1-scores of 84.8%, 51.1%, 61.3%, 74.6%, 91.0%, 56.3%, 59.3%, 98.8% and 70.8% on real-world datasets. And CNN gives the best result on Tales-Emotion dataset using FastText word embedding.

The convolution based emotion encoder of SENN is similar to the traditional CNN, but SENN emotion encoder only used the EWE emotion word embedding to extract emotion features. By concatenating semantic encoder, it improves the generalization of the model.

To the contrary, BiLSTM based semantic encoder of SENN is similar to the BiLSTM model. The SENN semantic encoder work with semantic word vectors and then the output is concatenated with emotion encoded vectors. It also generalizes both view of semantic and emotional information well. Depends on the characteristic of the datasets, size of the datasets, and vocabulary, the other deep learning based baseline models shows the comparative results. From the results, we can see that bidirectional GRU and bidirectional LSTM models outperformed the GRU and LSTM models respectively. And the combined RCNN and CNN-LSTM models work better than the other single architectures.

As shown in Table 4, we compared the execution time of the proposed model and the baseline models to process the tasks. However, deep learning algorithms are slower than machine learning algorithms, the accuracy is higher as shown in Table 3. The models CNN, GRU, BiGRU, LSTM, and BiLSTM are single subnetwork models which consist of

TABLE 4. Execution time comparison (seconds).

Model	D	C	T	TE	I	E	ET	G	EC	S
NB	1.55	2.37	1.93	0.52	0.56	0.74	0.28	0.23	0.23	0.04
RF	77.79	133.91	108.12	28.65	28.86	37.23	15.29	13.67	12.24	1.84
SVM	48.23	82.87	66.74	17.73	17.78	23.03	9.49	8.44	7.57	1.14
LR	29.74	51.20	41.15	10.96	11.03	14.25	5.84	5.23	4.65	0.70
CNN	45.25	77.90	63.01	16.66	16.78	21.75	8.91	7.97	7.12	1.07
GRU	56.61	97.41	78.66	20.86	20.92	27.10	11.12	9.95	8.93	1.34
BiGRU	63.67	109.59	88.73	23.50	23.61	30.43	12.53	11.71	10.01	1.51
LSTM	87.75	151.00	121.90	32.31	32.52	41.97	17.24	15.39	13.78	2.08
BiLSTM	90.64	155.78	126.07	33.34	33.57	43.43	17.79	15.90	14.26	2.15
RCNN	73.62	126.70	102.13	27.11	27.24	35.28	14.46	12.90	11.58	1.74
CNN LSTM	76.56	131.60	106.11	28.20	28.32	36.59	15.03	13.39	12.02	1.81
SENN	72.34	124.19	100.17	26.55	26.72	34.55	14.19	12.69	11.36	1.71

CNN or RNN subnetworks. Generally, the two subnetwork models are slower than single subnetwork models. The single LSTM and BiLSTM architectures execute the tasks slower than other compared models.

The proposed SENN model achieves the results in comparative execution time with RCNN and CNN LSTM two subnetwork architectures. The CNN architecture efficiently works in case of execution time.

From the above results, we selected FastText word embedding (also EWE word embedding for SENN model) and well-balanced dataset ISEAR for evaluating the impact of parameters in the next sub-section.

A. IMPACT OF PARAMETERS

We next evaluate the impacts of parameters of the proposed SENN model and compared with other deep learning based baseline models. In particular, we consider the following parameters: 1) the batch size; 2) the learning rate; 3) the dropout 4) the hidden size in RNN; 5) the number of layers in RNN; 6) the number of filters in CNN; 7) the filter sizes in CNN.

1) IMPACT OF BATCH SIZE

We first investigate the impact of the batch size of the SENN model and other deep learning based models. The batch size is an important parameter that influences the dynamics of the learning algorithm. We compared the several different batch size between 50 and 250. The performance of the proposed SENN model is constant on different size of the batch as shown in Figure 4. LSTM and GRU models are highly sensitive to batch size. It shows the low performance when changing the batch size. On the contrary, bidirectional LSTM and GRU shows the constant result on all experiments. LSTM and GRU perform good result when batch size is low such as 4, 8 and 16. It is, however, very slow during training. Generally, the proposed SENN model achieved the highest results on different size of the batch. The default batch size in this paper is 128.

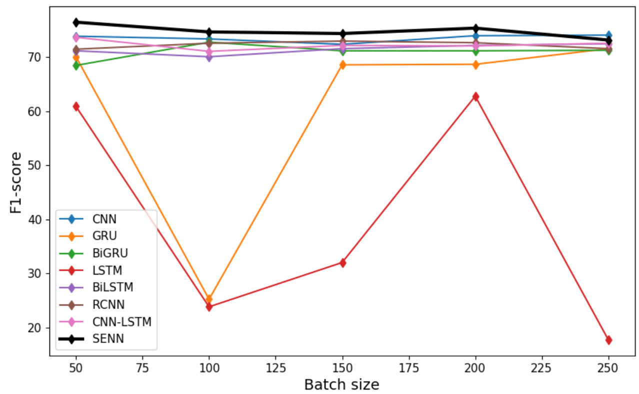

FIGURE 4. Relationship between batch size and F1 score.

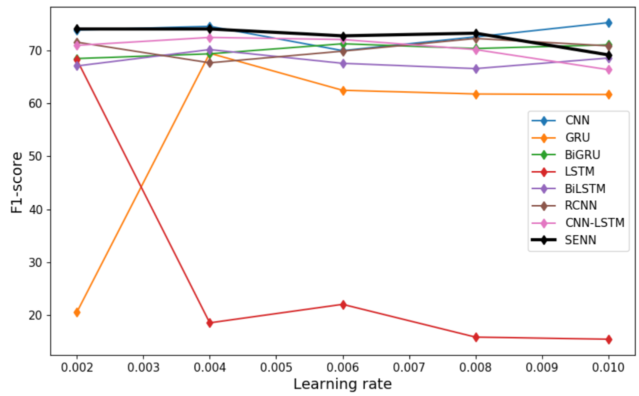

FIGURE 5. Relationship between learning rate and F1 score.

2) IMPACT OF LEARNING RATE

The appropriate choice of learning rate is important for the optimization of weights and offsets. If the learning rate is too large, it is easy to exceed the extreme point, making the system unstable. If the learning rate is too small, the training time is too long. Figure 5 shows the performance comparison of the proposed SENN model and other baseline models. SENN model achieved the highest result on a different configuration

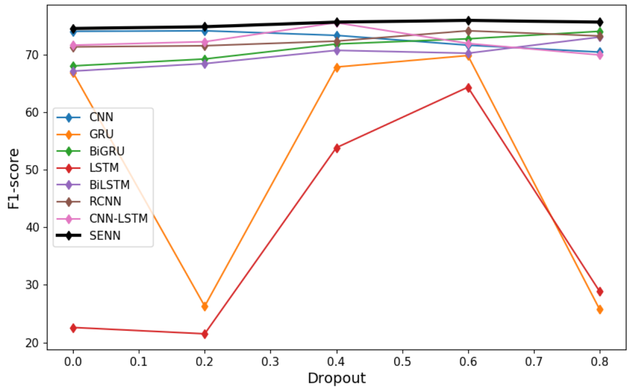

FIGURE 6. Relationship between dropout and F1 score.

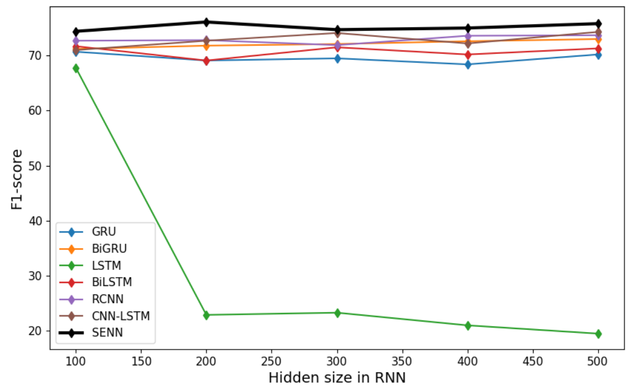

FIGURE 7. Relationship between the hidden size in RNN and F1 score.

of learning rate. When the learning rate is 0.01, the CNN model is performed better result than others. We considered the learning rates between values 0.002 and 0.01 for testing the impact of learning rate. LSTM and GRU models are also very sensitive to different learning rates. We set the default learning rate in this paper as 0.001.

3) IMPACT OF DROPOUT

In order to prevent the over-fitting phenomenon in the training process, the dropout mechanism was introduced, and our SENN model performs the best result when dropout is changed. Figure 6 is the relationship between dropout and F1 score. We tested the models with different size of dropout between 0.0 and 1.0. SENN model has two different dropouts for each sub-network, the one is at the end of CNN emotion encoder, and the other is at the end of BiLSTM semantic encoder. On the comparison, SENN model performs better result than other baseline models. We used the same value for both sub-networks to evaluate the impact of dropout. The default dropout in this paper is 0.5.

4) IMPACT OF HIDDEN SIZE IN RNN

The number of hidden layer nodes influences the complexity and effect of the method. If the number of nodes is too small, the network learning ability will be limited. If the number of nodes is too large, the complexity of the network structure is large. At the same time, it is easier to fall into local minimum points during the training process, and the network learning speed will decrease. Figure 7 shows the comparison of models in case of a different number of hidden layer in RNN. We tested the different hidden sizes between 100 and 500. The proposed SENN model achieved a better result in all experiments. We set the default number of the hidden layer as 256.

5) IMPACT OF NUMBER OF LAYERS IN RNN

It can be seen from Figure 8 that the proposed SENN model performs better results constantly compared with other baseline models. We tested the impact of a number of layers on

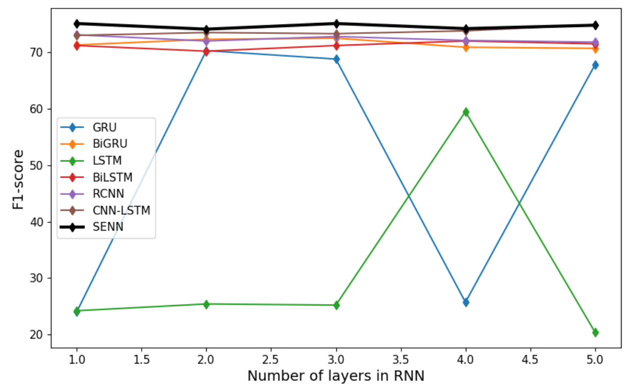

FIGURE 8. Relationship between a number of layers in RNN and F1 score.

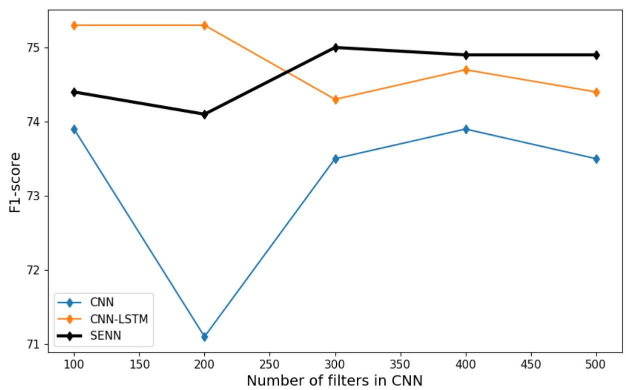

FIGURE 9. Relationship between a number of filters in CNN and F1 score.

different values between 1 and 5. We set the default hidden layer size in this paper as 2. LSTM and GRU models also perform low F1 score on all experiments.

6) IMPACT OF NUMBER OF FILTERS IN CNN

We next investigate the impact of hidden layer size. In particular, we first fix the hidden layer size between 100 and 500. Figure 9 shows the relationship between the number of filters and F1 score. The experimental result shows that when increasing the number of filters, the proposed SENN model

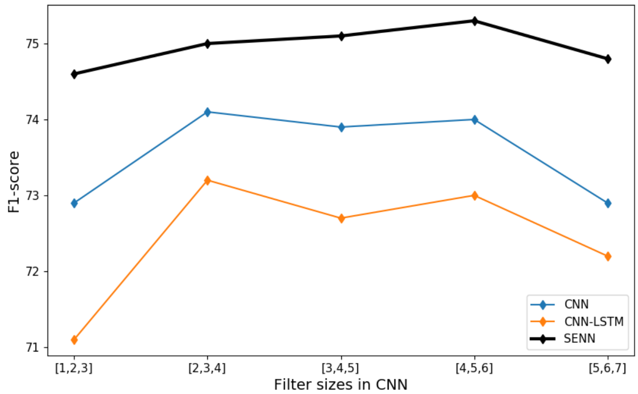

FIGURE 10. **Relationship between filter sizes in CNN and F1 score.**

performs the highest results. When a number of filters are 100 and 200, the combined CNN-LSTM performs higher result than the other models. We set the default number of filters in this paper as 100.

7) IMPACT OF FILTER SIZES IN CNN

It can be seen from Figure 10 that when filter size is set of 4, 5, and 6, the SENN model achieved the highest F1 score of 75.3%. We tested the different set of filter sizes such as [1,2,3], [2,3,4], [3,4,5], [4,5,6] and [5,6,7]. The proposed SENN model performs a better result on all experiments. The default filter sizes used are set of 3, 4 and 5 in the comparison with other methods. The tradition CNN models give a more comparative result with SENN model than the other model.

In all comparison of models, RNN models such as LSTM and GRU is highly sensitive because of batch size on GPUs. Performance at lower batches is especially important when using data parallelism to distribute the computation across GPUs. When batch size is big enough, this will provide a stable enough estimate of average gradient of the full dataset.

VI. CONCLUSION

In the era of the rapid development of social network and internet of things, it is very meaningful to explore the emotional recognition of user-generated data through artificial intelligence technology. This paper explored an emotion recognition method from text based on the combined network which consists of CNN based emotion encoder and BiLSTM based semantic encoder called SENN, a novel model is proposed and applied on ten real-world datasets. For the SENN model, BiLSTM is designed to capture contextual information and CNN is designed to extract emotional information effectively. In the experimental work, we have conducted the ten datasets and then analyzed using the proposed SENN model the other baseline models including state-of-the-art machine learning and deep learning models. Experiments on emotionally related datasets show that our method can achieve better performances compared with state-of-the-art baseline methods. Our proposed framework is general enough to be applied to more scenarios. In the future works, we will extend proposed CNN based emotion encoding and BiLSTM based semantic encoding to other tasks such as affective computing and sentiment analysis. It is possible to improve the performance of SENN model by using the larger emotion word embeddings.

REFERENCES

[1] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Process. Mag.*, vol. 18, no. 1, pp. 32–80, Jan. 2001.

[2] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer, "The first facial expression recognition and analysis challenge," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Workshops (FG)*, Mar. 2011, pp. 921–926.

[3] M. Karg, A.-A. Samadani, R. Gorbet, K. Kuhnlenz, J. Hoey, and D. Kulic, "Body movements for affective expression: A survey of automatic recognition and generation," *IEEE Trans. Affective Comput.*, vol. 4, no. 4, pp. 341–359, Oct. 2013.

[4] M. Soleymani, S. Koelstra, I. Patras, and T. Pun, "Continuous emotion detection in response to music videos," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Workshops (FG)*, Mar. 2011, pp. 803–808.

[5] F. Agrafioti, D. Hatzinakos, and A. K. Anderson, "ECG pattern analysis for emotion detection," *IEEE Trans. Affective Comput.*, vol. 3, no. 1, pp. 102–115, Jan./Mar. 2012.

[6] P. Rodriguez, A. Ortigosa, and R. M. Carro, "Extracting emotions from texts in e-learning environments," in *Proc. 6th Int. Conf. Complex, Intell., Softw. Intensive Syst.*, Jul. 2012, pp. 887–892.

[7] A. Jamshidnejad and A. Jamshidined, "Facial emotion recognition for human computer interaction using a fuzzy model in the e-business," in *Proc. Innov. Technol. Intell. Syst. Ind. Appl. (CITISIA)*, Jul. 2009, pp. 202–204.

[8] S. Leek and G. Christodoulides, "A framework of brand value in B2B markets: The contributing role of functional and emotional components," *Ind. Marketing Manage.*, vol. 41, no. 1, pp. 106–114, Jan. 2012.

[9] P. Ekman, "An argument for basic emotions," *Cognition Emotion*, vol. 6, nos. 3–4, pp. 169–200, May 1992.

[10] R. Plutchik, "A general psychoevolutionary theory of emotion," *Theories Emotion*, vol. 1 no. 3, pp. 3–33, 1980.

[11] 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, Jul./Aug. 2018, Art. no. e1253.

[12] Y. Hao, Y. Zhang, K. Liu, S. He, Z. Liu, H. Wu, and J. Zhao, "An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Vol. 1, Jul. 2017, pp. 221–231.

[13] Y. Y. H. Ding Liu Luan and M. Sun, "Visualizing and understanding neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2017, pp. 1150–1159.

[14] T. Mikolov, I. C. K. Sutskever, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[15] 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.

[16] P. Xu, A. Madotto, C. S. Wu, J. H. Park, and P. Fung, "Emo2vec: Learning generalized emotion representation by multi-task training," Sep. 2018, *arXiv:1809.04505*. [Online]. Available: https://arxiv.org/abs/1809.04505

[17] A. Agrawal, A. An, and M. Papagelis, "Learning emotion-enriched word representations," in *Proc. 27th Int. Conf. Comput. Linguistics*, Aug. 2018, pp. 950–961.

[18] S. Buechel and U. Hahn, "Emotion representation mapping for automatic lexicon construction (mostly) performs on human level," Jun. 2018, *arXiv:1806.08890*. [Online]. Available: https://arxiv.org/abs/1806.08890

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

[20] M. Abdul-Mageed and L. Ungar, "Emonet: Fine-grained emotion detection with gated recurrent neural networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 4, Jul. 2017, pp. 718–728.

[21] N. Colneriĉ and J. Demsar, "Emotion recognition on Twitter: Comparative study and training a unison model," *IEEE Trans. Affective Comput.*, to be published.

[22] B. Kratzwald, S. Ilić, M. Kraus, S. Feuerriegel, and H. Prendinger, "Deep learning for affective computing: Text-based emotion recognition in decision support," *Decis. Support Syst.*, vol. 115, pp. 24–35, Nov. 2018.

[23] S. M. Mohammad, "From once upon a time to happily ever after: Tracking emotions in mail and books," *Decis. Support Syst.*, vol. 53, no. 4, pp. 730–741, Nov. 2012.

[24] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word–emotion association lexicon," *Comput. Intell.*, vol. 29, no. 3, pp. 436–465, Aug. 2013.

[25] C. Strapparava and A. Valitutti, "Wordnet affect: An affective extension of WordNet," in *Proc. LREC*, vol. 4, May 2004, p. 40.

[26] G. Miller, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press. 1998.

[27] J. Staiano and M. Guerini, "Depechemood: A lexicon for emotion analysis from crowd-annotated news," May 2014, *arXiv:1405.1605*. [Online]. Available: https://arxiv.org/abs/1405.1605

[28] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining Text Data*. Boston, MA, USA: Springer, 2012, pp. 415–463.

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

[30] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: LIWC 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001. 2001.

[31] M. M. Bradley and P. J. Lang, "Affective norms for english words (ANEW): Instruction manual and affective ratings,"Center Res. Psychophysiol., Univ. Florida, Gainesville, FL, USA, Tech. Rep. C-1, 1999. Vol. 30, no. 1, pp. 25–36.

[32] A. Agrawal and A. An, "Unsupervised emotion detection from text using semantic and syntactic relations," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, vol. 1, Dec. 2012, pp. 346–353.

[33] T. Danisman and A. Alpkocak, "Feeler: Emotion classification of text using vector space model," in *Proc. AISB Conv. Commun., Interact. Social Intell.*, vol. 1, Apr. 2008, p. 53.

[34] S. Chaffar and D. Inkpen, "Using a heterogeneous dataset for emotion analysis in text," in *Proc. Canadian Conf. Artif. Intell.* Berlin, Germany: Springer, 2011, pp. 62–67.

[35] M. Hasan, E. Rundensteiner, and E. Agu, "Emotex: Detecting emotions in Twitter messages," ASE BIGDATA/SOCIALCOM/CYBERSECURITY Conf., Stanford, CA, USA, Tech. Rep., 2014.

[36] D. Gordeev, "Detecting state of aggression in sentences using CNN," in *Proc. Int. Conf. Speech Comput.* Cham, Switzerland: Springer, 2016, pp. 240–245.

[37] A. Seyeditabari, S. Levens, C. D. Maestas, S. Shaikh, J. I. Walsh, W. Zadrozny, C. Danis, and O. P. Thompson, "Cross corpus emotion classification using survey data," presented at the AISB, 2017.

[38] D. Chatzakou, A. Vakali, and K. Kafetsios, "Detecting variation of emotions in online activities," *Expert Syst. Appl.*, vol. 89, pp. 318–332, Dec. 2017.

[39] M. Purver and S. Battersby, "Experimenting with distant supervision for emotion classification," in *Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Apr. 2012, pp. 482–491.

[40] S. Wen and X. Wan, "Emotion classification in microblog texts using class sequential rules," in *Proc. 28th AAAI Conf. Artif. Intell.*, Jun. 2014, pp. 187–193.

[41] A. Bandhakavi, N. Wiratunga, D. Padmanabhan, and S. Massie, "Lexicon based feature extraction for emotion text classification," *Pattern Recognit. Lett.*, vol. 93, pp. 133–142, Jul. 2017.

[42] C. O. Alm, D. Roth, and R. Sproat, "Emotions from text: Machine learning for text-based emotion prediction," in *Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process.* Oct. 2005, pp. 579–586.

[43] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proc. 4th Int. Workshop Semantic Eval. (SemEva)*, Jun. 2007, pp. 70–74.

[44] B. Schölkopf, J. Weston, E. Eskin, C. Leslie, and W. S. Noble, "A kernel approach for learning from almost orthogonal patterns," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, 2002, pp. 511–528.

[45] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proc. 23rd Int. Conf. Mach. Learn.*, Jun. 2006, pp. 377–384.

[46] U. Gupta, A. Chatterjee, R. Srikanth, and P. Agrawal, "A sentiment-and-semantics-based approach for emotion detection in textual conversations," Jul. 2017, *arXiv:1707.06996*. [Online]. Available: https://arxiv.org/abs/1707.06996

[47] H. Ren, J. Wan, and Y. Ren, "Emotion detection in cross-lingual text based on bidirectional LSTM," in *Proc. Int. Conf. Secur. Intell. Comput. Big-Data Services*. Cham, Switzerland: Springer, 2018, pp. 838–845.

[48] A. Illendula and A. Sheth, "Multimodal emotion classification," Mar. 2019, *arXiv:1903.12520*. [Online]. Available: https://arxiv.org/abs/1903.12520

[49] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[50] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Černocký, "Empirical evaluation and combination of advanced language modeling techniques," in *Proc. 12th Annu. Conf. Int. Speech Commun. Assoc.*, Aug. 2011, pp. 27–31.

[51] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," Feb. 2018, *arXiv:1802.05365*. [Online]. Available: https://arxiv.org/abs/1802.05365

[52] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Oct. 2018, *arXiv:1810.04805*. [Online]. Available: https://arxiv.org/abs/1810.04805

[53] P. Zhong and C. Miao, "Ntuer at SemEval-2019 Task 3: Emotion classification with word and sentence representations in RCNN," Feb. 2019, *arXiv:1902.07867*. [Online]. Available: https://arxiv.org/abs/1902.07867

[54] L.-A.-M. Bostan and R. Klinger, "An analysis of annotated corpora for emotion classification in text," in *Proc. 27th Int. Conf. Comput. Linguistics*, Aug. 2018, pp. 2104–2119.

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

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

[57] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A C-LSTM neural network for text classification," Nov. 2015, *arXiv:1511.08630*. [Online]. Available: https://arxiv.org/abs/1511.08630

[58] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "Dailydialog: A manually labelled multi-turn dialogue dataset," Oct. 2017, *arXiv:1710.03957*. [Online]. Available: https://arxiv.org/abs/1710.03957

[59] S. Mohammad and S. Kiritchenko, "Understanding emotions: A dataset of tweets to study interactions between affect categories," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018, pp. 198–209.

[60] E. Agirre, J. Bos, S. Manandhar, Y. Marton, and D. Yuret, "SEM 2012: The first joint conference on lexical and computational semantics," in *Proc. Main Conference Shared Task, 6th International Workshop Semantic Evaluation (SemEval 2012)*, vols. 1–2, 2012, pp. 1–32.

[61] K. R. Scherer and H. G. Wallbott, "Evidence for universality and cultural variation of differential emotion response patterning," *J. Personality Social Psychol.*, vol. 66, no. 2, pp. 310–328, 1994.

[62] S. M. Mohammad and F. Bravo-Marquez, "Emotion intensities in tweets," Aug. 2017, *arXiv:1708.03696*. [Online]. Available: https://arxiv.org/abs/1708.03696

[63] S. M. Mohammad and S. Kiritchenko, "Using hashtags to capture fine emotion categories from tweets," *Comput. Intell.*, vol. 31, no. 2, pp. 301–326 May 2015.

[64] A. Gelbukh, "Computational linguistics and intelligent text processing," in *Proc. 16th Int. Conf.*, vol. 9042. Cairo, Egypt, Springer, 2015, pp. 14–20.

[65] H. Schuff, J. Barnes, J. Mohme, S. Padó, and R. Klinger, "Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus," in *Proc. 8th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, Sep. 2017, pp. 13–23.

ERDENEBILEG BATBAATAR received the B.S. degree in software engineering from the National University of Mongolia, in 2013, and the M.S. and Ph.D. degrees in computer science from Chungbuk National University, Cheongju, South Korea, in 2019, where he is currently holds a Postdoctoral position. His research interests include software engineering, data mining, big data analysis, bioinformatics, machine learning, and deep learning.

MEIJING LI received the B.S. degree in computer science from Dalian University, Dalian, China, in 2007, and the M.S. degree in bio information technology and the Ph.D. degree in computer science from Chungbuk National University, Cheongju, South Korea, in 2010 and 2015, respectively, where she held a Postdoctoral position. She is currently an Assistant Professor with the College of Information Engineering, Shanghai Maritime University, Shanghai, China. Her research interests include data mining, information retrieval, database systems, bioinformatics, and biomedicine.

KEUN HO RYU (LM'83) received the Ph.D. degree in computer science and engineering from Yonsei University, South Korea, in 1988. He has served at the Reserve Officers' Training Corps (ROTC) of the Korean Army. He was with The University of Arizona, Tucson, AZ, USA, as a Postdoctoral and a Research Scientist, and also with the Electronics and Telecommunications Research Institute, South Korea, as a Senior Researcher. He is currently a Professor with the Faculty of Information Technology, Ton Duc Thang University, Vietnam, as well as an Emeritus and Invited Professor with Chungbuk National University, South Korea. He is also an Honorary Doctorate of the National University of Mongolia. He has been the Leader of the Database and Bioinformatics Laboratory, South Korea, since 1986. He is the former Vice-President of the Personalized Tumor Engineering Research Center.

He has published or presented over 1000 referred technical articles in various journals and international conferences, in addition to authoring a number of books. His research interests include temporal databases, spatiotemporal databases, temporal GIS, stream data processing, knowledge-based information retrieval, data mining, biomedical informatics, and bioinformatics. He has been a member of the ACM, since 1983. He has served on numerous program committees, including roles as the Demonstration Co-Chair of the VLDB, as the Panel and Tutorial Co-Chair of the APWeb, and as the FITAT General Co-Chair.