Convolutional Transformer with Sentiment-aware Attention for Sentiment Analysis

Pengfei Li
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
pli006@e.ntu.edu.sg

Peixiang Zhong
School of Computer Science and Engineering
Nanyang Technological University
Singapore
peixiang001@e.ntu.edu.sg

Jiaheng Zhang
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
e160003@e.ntu.edu.sg

Kezhi Mao*
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
ekzmao@ntu.edu.sg

***Abstract*—Given certain data available for training, the keys to improving a sentiment analysis system lie in developing a good model that is capable of capturing both local and global features of texts, as well as incorporating external knowledge into the model effectively. In this paper, we propose a multi-window Convolutional Transformer (ConvTransformer) that takes the advantages of both Transformer and CNN for sentiment analysis. The proposed ConvTransformer is able to capture important local n-gram features effectively while preserving sequential information of texts. Furthermore, we propose a sentiment-aware attention mechanism to incorporate the sentiment intensity information of each word by utilizing an external knowledge base, SentiWordNet. The sentiment-aware attention mechanism takes both sentiment and position information of each token into consideration when computing attention weights, resulting in a global feature for final classification. Comparing with CNN, RNN and attention-based baseline models, our model achieves the best performance on multiple sentiment analysis datasets.**

***Index Terms*—sentiment analysis, Transformer, CNN, SentiWordNet, attention**

I. Introduction

Nowadays, people use Internet extensively to read or post articles, reviews and comments expressing opinions towards certain product, event or topic. Understanding the sentiment of customers is crucial for businesses and organizations to review their products, policies or business strategies. The overwhelming of such text data on the Internet has led to the revolution of automatically analyzing and extracting the sentiment from text. Sentiment analysis is a Natural Language Processing (NLP) technique that automatically identifies and categorizes opinions expressed in a piece of text, especially to determine its sentiment polarity (e.g. positive, negative, or neutral).

Early works on sentiment analysis focus on lexicon and rule-based approaches [1]–[4], which rely on a large set of handcrafted rules and extensive human effort to construct and maintain the rules. Traditional machine learning approaches classify sentiment polarity based on rich features (lexical, syntactic or semantic features) using classifiers such as Naive Bayes and SVM [5]–[8]. However, it requires sophisticated feature engineering and the performance strongly depends on the quality of the designed features. Recently, deep learning approaches become more and more popular since it is able to extract meaningful features automatically and more effectively by leveraging deep neural networks.

The commonly adopted deep neural network architectures for sentiment analysis include Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Transformer, each architecture has its own advantages and limitations. CNN is a special feed-forward neural network with convolutional layers interleaved with pooling layers. In NLP, 1d convolution is normally performed to compute the semantic relevance between the n-grams in text and convolutional filters/kernels learned by the network. With the help of pooling operations, the important n-grams appear in the text can be captured. Hence, CNN is able to capture local features effectively and induce more abstract and informative representation for sentiment analysis [9]–[11]. However, conventional CNN is relatively weak in capturing sequential dependencies since the pooling operation ignores the position information. RNN is well-known for processing sequences of inputs while updating its internal states recurrently. Intuitively, it is well suited for sentiment analysis since text data is also a type of sequential data [12]–[14]. However, RNN suffers from gradient vanishing and parallel-unfriendly problems due to its recurrent nature. A recent proposed neural architecture called Transformer [15] addresses the problems of RNN by relying entirely on self-attention for text representation. It has shown to be more effective than RNN for sentiment analysis [16]–[18] and many other NLP tasks [19], [20]. However, Transformer has limited ability in capturing local n-gram features which are crucial for sentiment analysis since self-attention focuses on global

* Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

instead of local context. Besides, the heavy architecture of Transformer often requires more training data, computational power and memory, especially for long texts.

To address the limitations of above-mentioned neural architectures, we propose a multi-window Convolutional Transformer (ConvTransformer) which takes the advantages of both CNN and Transformer for efficient text representation. ConvTransformer has a simplified multi-head multi-layer structure compared with Transformer. In each head, instead of self-attention, convolution operation over the text represented in different sub-spaces is performed to better capture local n-gram features. To effectively capture the n-grams with various lengths, we use multiple convolution window sizes in different convolutional heads, allowing different heads to focus on different n-gram features. We also discard the pooling operation in order to preserve sequential information in the output of ConvTransformer. Moreover, we propose a sentiment-aware attention mechanism to summarize the sequential output of ConvTransformer and obtain the global text representation by incorporating external knowledge of words' sentiment from SentiWordNet. The sentiment intensity information used in sentiment-aware attention is beneficial for sentiment analysis and also reduces model's reliance on training data to learn everything from scratch. Meanwhile, position information is also considered when calculating attention weights.

The contributions of this paper are summarized as follows: (1) We propose a novel neural architecture called Convolutional Transformer (ConvTransformer), which is able to capture n-grams features effectively while preserving sequential information; (2) We propose a sentiment-aware attention mechanism to effectively incorporate sentiment intensity information of words from an external knowledge base into deep neural networks; (3) Comprehensive experiments and analyses show that our proposed model outperforms CNN, RNN and attention-based baseline models for sentiment analysis.

II. Related Work

We focus here on deep learning approaches for sentiment analysis since they have demonstrated better performance than rule-based and traditional machine learning approaches.

Convolutional Neural Network (CNN) is originally applied in the field of computer vision for image feature extraction [21]. With the development of distributed representations of words in NLP [22], many CNN-based models using word embeddings as input were proposed to learn the representation of texts for sentiment analysis. Kim [9] first used CNN for sentence classification. Dynamic CNN (DCNN) was also proposed for sentence modeling which uses a dynamic K-Max pooling as non-linear sub-sampling function [23]. In [10], character-level CNN was proposed which uses character embeddings as input instead of word embeddings. In [11], Character to Sentence CNN (CharSCNN) was proposed which jointly uses character-level, word-level and sentence-level representations for sentiment analysis. Deep CNNs consisting of many convolutional and pooling layers also achieved good performance on sentiment analysis tasks [24], [25].

Besides CNN-based models, Recurrent Neural Networks (RNNs) are also widely used for sentiment analysis. RNNs are suitable for modeling sequential inputs like texts since the network is conditioned on all previously seen inputs and propagating the effects of words over the sentence recurrently. Practically, the improved variants of RNN such as Long Short-Term Memory Neural Network (LSTM) [12]–[14], Gated Recurrent Neural Network (GRU) [26], bi-directional LSTM (BiLSTM) [27] and Disconnected Recurrent Neural Network (DRNN) [28] are used for sentiment analysis to alleviate the long distance gradient vanishing problem. Some works combined CNN and RNN as hybrid models to utilize the benefits of different neural architectures [29], [30] .

To solve the gradient vanishing problem and better capture long-distance dependencies, attention mechanisms are widely used in deep neural networks [31]. Many works focused on improving CNN and RNN with attention mechanisms. In [32], hierarchical attention network (HAN) was proposed for document-level sentiment analysis, which uses two levels of attention mechanisms to find the key structural information of the document. In [33], attention-based LSTM was proposed for aspect-level sentiment analysis, in which attention mechanism was used to concentrate on different parts of the sentence when different aspects are taken as input. Reference [34] introduced a matrix sentence embedding representation using BiLSTM with self-attention mechanism. In [35], three different attentions including attention vector, LSTM attention and attentive pooling were integrated with CNN model.

Recently, a novel neural architecture called Transformer was proposed which is based solely on the self-attention mechanism for text representation [15]. Since self-attention is able to draw global dependencies between input and output without regard to their distances in the sequence, Transformer is more suitable for capturing long-distance dependencies and allows more parallelization compared with RNN. The emerging of Transformer has led to a series of breakthroughs in a wide range of NLP tasks including sentiment analysis [16]–[18]. Especially, the pre-trained language models based on Transformer have achieved state-of-the-art performance on many benchmark datasets [36], [37].

Deep learning approaches normally require large amount of training data to achieve good performance. Data scarcity or low quality of training data will lead to bad generalization. Many works focused on improving text representation by incorporating external knowledge into deep neural networks [19], [38], [39]. For sentiment analysis, many emotional words and sentiment terms are critical to determine the whole sentiment of a sentence. Many works utilized such sentiment knowledge from external knowledge bases such as SentiWordNet [40], [41], SenticNet [42] and ConceptNet [20] to improve the performance of sentiment analysis.

Different from existing approaches, our model combines CNN and Transformer to take the advantages of both neural architectures for efficient text representation. We also incorporate external knowledge into our model in a novel and effective way to better capture words' sentiment intensity information.

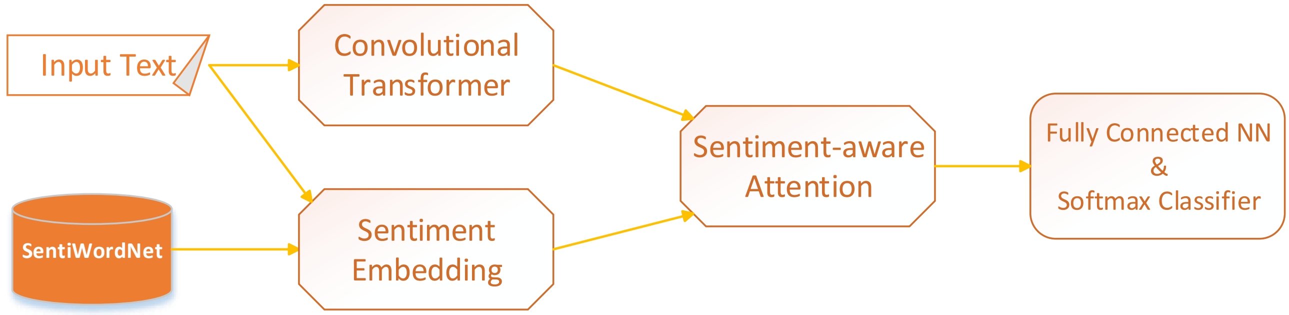

Fig. 1: Overall architecture of the proposed sentiment analysis system.

III. Methodology

In this chapter, we present our methodologies in detail. The overall architecture of our model is shown in Fig. 1. The input text is encoded using the proposed Convolutional Transformer (ConvTransformer) to capture local n-gram features (Section III-A); Meanwhile, sentiment embedding is generated for each word in the input text utilizing external knowledge base (Section III-B); Then, sentiment-aware attention is used to summarize the output of ConvTransformer and obtain the final representation by incorporating sentiment intensity information and position information of each token (Section III-C); Finally, sentiment polarity of the text is classified based on the final representation (Section III-D).

A. Convolutional Transformer Encoder

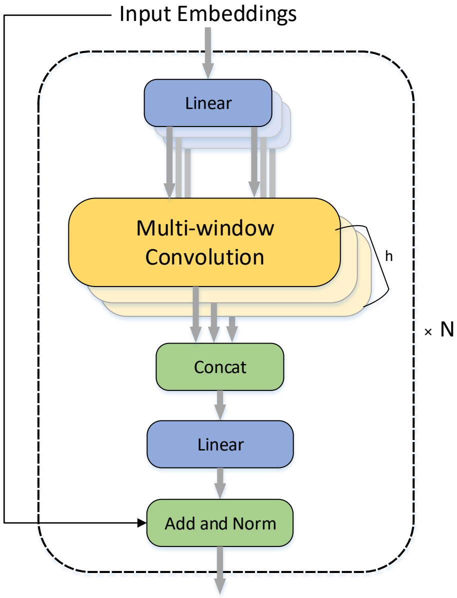

Fig. 2: Convolutional Transformer (ConvTransformer).

ConvTransformer is the main component of our sentiment analysis system, which encodes text using multiple convolutional and linear operations. It is able to capture important local n-gram features for sentiment analysis while preserving sequential information. The architecture of the proposed ConvTransformer is shown in Fig. 2.

1) N-gram Convolution Operation: Convolution is the fundamental operation of ConvTransformer which computes the semantic relevance between n-grams in the text and trainable convolutional filters. Specifically, given the input text $t = [t_1, t_2, ..., t_l]$, each word t_i is represented as d_w-dimensional word embedding $\mathbf{x}_i$ by looking up the word embedding matrix $\mathbf{W}^{wrd} \in \mathbb{R}^{d_w \times V}$, where V is vocabulary size. Then, n-gram convolution over input embeddings $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l]^T$ is performed using convolutional filters $\mathbf{F} = [\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_m}]^T$, where $\mathbf{f_i} \in \mathbb{R}^{nd_w}$ and n is the convolution window size. Consequently, a feature map $\mathbf{M} \in \mathbb{R}^{l \times m}$ is generated as follows:

$$\mathbf{M} = \mathbf{X} \circledast \mathbf{F}^T \tag{1}$$

where $\circledast$ indicates the convolution operation of $\mathbf{f_i}$ over $\mathbf{X}$. Specifically, the value in the feature map is calculated as shown in Equation 2:

$$m_{ji} = f(\mathbf{f_i}^T \cdot \text{Cat}(x_j, x_{j+1}, ..., x_{j+n-1}) + b) \tag{2}$$

where Cat means concatenation, f is a non-linear activation function and b is a bias term.

2) Multi-head Multi-layer Structure: Inspired by the structure of Transformer [15], ConvTransformer also has multi-head multi-layer structure that allows the model to extract features in both parallel and sequential manners. As shown in Fig. 2, for a h-head ConvTransformer, the input embedding $\mathbf{X}$ is first transformed into h sub-spaces using different linear transforms. Then, different n-gram convolution is performed in different sub-spaces using multiple convolution window sizes. Finally, the outputs from all the convolutional heads are concatenated together and linearly transformed to the original input dimension, as shown in Equation 3.

$$\begin{gathered}\text{ConvTransformer}(\mathbf{X}) = \text{Cat}(\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_h)\mathbf{W}^M \\ \text{where } \mathbf{M}_i = \text{Conv}(\mathbf{X}\mathbf{W}_i^X)\end{gathered} \tag{3}$$

Here, Cat indicates column-wise concatenation, Conv indicates n-gram convolution operation, $\mathbf{W}_i^X \in \mathbb{R}^{d_w \times (d_w/h)}$ and $\mathbf{W}^M \in \mathbb{R}^{hm \times d_w}$ are the weight matrices of linear transformations. Moreover, we adopt the residual connection and layer norm as used in [15].

Compared with conventional CNN where the convolution operation is performed on words directly, ConvTransformer performs convolution in different sub-spaces of words simultaneously, where the words in different sub-spaces may represent different meanings. Such multi-head structure allows our model to capture more semantic or syntactic features from n-grams and gives our model more capabilities in dealing with complex NLP tasks like sentiment analysis. Besides, various convolution window sizes are used in different convolutional heads, allowing ConvTransformer to capture n-grams with various lengths effectively while each head focuses on specific n-grams. We also discard the pooling operation that is normally used in conventional CNNs, making ConvTransformer a sequence-to-sequence model. Therefore, the sequential information of texts is preserved.

Compared with Transformer which uses self-attention as its fundamental operation, our ConvTransformer uses convolution that focuses more on local instead of global context. This allows our model to capture local n-gram features more effectively than Transformer. These n-gram features are important keywords or phrases that are crucial for sentiment analysis. Besides, ConvTransformer has more simplified structure, hence it is more memory and computational efficient than Transformer, especially for long texts.

B. Sentiment Embedding Generation

In this section, we introduce our methodology of utilizing the external knowledge about words' sentiment and generating sentiment embedding that can be incorporated into our model effectively.

SentiWordNet[1] is a publicly available and frequently used lexical knowledge base for sentiment analysis and opinion mining [43], [44]. It is constructed based on WordNet [45], a large lexical database that groups English words into synonym sets (synsets) to represent different meanings or concepts. SentiWordNet assigns each synset a positivity score and a negativity score ranging from 0 to 1 to reflect its sentiment intensity. Table I shows some examples of SentiWordNet lexicons. Such knowledge of words' sentiment intensity can be utilized as prior knowledge in our model to improve sentiment analysis performance.

Specifically, for each word in the vocabulary, we search SentiWordNet for its positivity and negativity scores and then transform the scores into integers ranging from 1 to 10 to reflect its sentiment intensity, as shown in Equation 4.

$$I = \begin{cases} \lfloor 10 \times score \rfloor + 1 & \text{if } score \neq 1 \\ 10 & \text{if } score = 1 \end{cases} \tag{4}$$

where $score$ represents positivity or negativity score ranging from 0 to 1. For unknown word that is not in SentiWordNet, we set both its positivity and negativity scores into 0 which means neutral or objective word.

TABLE I: Example words and their sentiment intensities obtained from SentiWordNet. The (POS,ID) pair uniquely identifies a WordNet synset; *PosScore* and *NegScore* represent positivity score and negativity score respectively.

Word	**(POS, ID)**	***PosScore***	***NegScore***
excellent	(a, 02343110)	1	0
miserable	(a, 01150205)	0	0.875
gentle	(a, 01455412)	0.375	0.25
employer	(n, 10054657)	0	0

Given the input text $t = [t_1, t_2, ..., t_l]$, we are able to find the positive sentiment intensity I_i^p and negative sentiment intensity I_i^n of each word t_i. Then we transform the sentiment intensities into sentiment embeddings $\mathbf{s}_i^p$ and $\mathbf{s}_i^n$ by constructing two sentiment embedding matrices $\mathbf{W}^{pos}, \mathbf{W}^{neg} \in \mathbb{R}^{d_s \times 11}$, one for positive and the other for negative sentiment embedding. Here, d_s is the dimension of sentiment embedding, 11 means ten embeddings for the ten sentiment intensity levels and one embedding for padding tokens[2]. The final sentiment embedding for t_i is obtained by concatenating the positive and negative sentiment embeddings: $\mathbf{s}_i = \text{Cat}(\mathbf{s}_i^p, \mathbf{s}_i^n)$.

C. Sentiment-aware Attention

To effectively incorporate the sentiment intensity information of sentiment embedding and obtain the final text representation, we propose a sentiment-aware attention mechanism that summarizes the output of ConvTransformer while taking both sentiment and position information into consideration. As shown in Fig. 3, the attention weight of each token α_i is calculated based on its sentiment embedding $\mathbf{s}_i$ and position embedding $\mathbf{p}_i$ besides the ConvTransformer output $\mathbf{o}_i$. The position embedding $\mathbf{p}_i$ is obtained by mapping the token's absolute position to d_p-dimensional embeddings based on a trainable position embedding matrix $\mathbf{W}^p \in \mathbb{R}^{d_p \times P}$, where P is the total number of positions.

Formally, the attention weight α_i is calculated as follows:

$$u_i = \mathbf{c}^{\mathrm{T}}\tanh(\mathbf{W}_o\mathbf{o}_i + \mathbf{W}_s\mathbf{s}_i + \mathbf{W}_p\mathbf{p}_i) \tag{5}$$

$$\alpha_i = \frac{exp(u_i)}{\sum_{j=1}^{l} exp(u_j)} \tag{6}$$

where $\mathbf{W}_o \in \mathbb{R}^{d_a \times d_w}$, $\mathbf{W}_s \in \mathbb{R}^{d_a \times 2d_s}$, $\mathbf{W}_p \in \mathbb{R}^{d_a \times d_p}$, d_a is attention dimension, and $\mathbf{c} \in \mathbb{R}^{d_a}$ is a context vector learned by the neural network. The attention weight α_i reflects the contribution of each token to the final text representation, and the final representation is computed as shown in Equation 7.

$$\mathbf{f} = \sum_{i=1}^{l} \alpha_i \mathbf{o}_i \tag{7}$$

The sentiment embedding used in our sentiment-aware attention mechanism contains prior knowledge of words' sentiment intensity, it is beneficial for sentiment analysis and also

[1] Available for download from https://github.com/aesuli/sentiwordnet.

[2] We use fixed zero vector as the sentiment embedding of padding tokens.

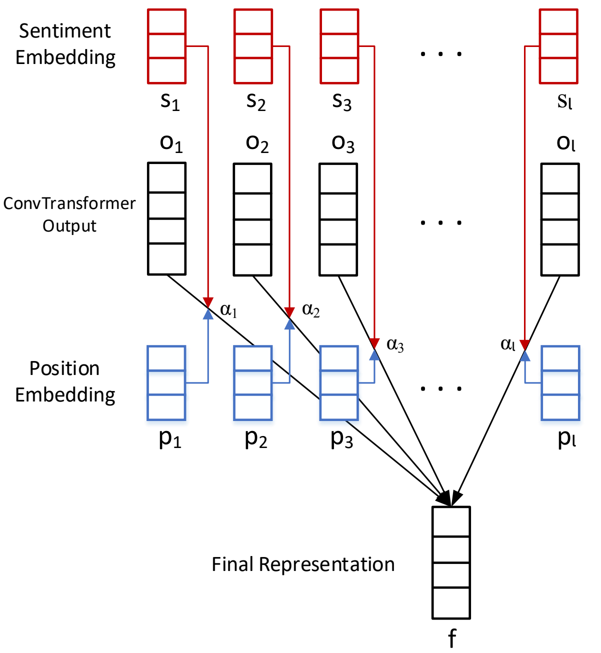

Fig. 3: Sentiment-aware attention mechanism. Attention weights α_i are calculated based on ConvTransformer output $\mathbf{o}_i$, sentiment embedding $\mathbf{s}_i$ and position embedding $\mathbf{p}_i$ of each token.

reduces the reliance on training data; The position embedding used in sentiment-aware attention provides positional information of each token, which is important for sentiment analysis since words appear in different positions may have different affective meanings.

D. Classification and Optimization

After obtaining the final text representation $\mathbf{f}$, we pass it to a classifier to compute the probability for each sentiment class. The classifier consists of a fully connected layer of standard neural network for dimension reduction and a softmax layer (fully connected layer with softmax activation function) for class probabilities calculation. For optimization, we train our model by minimizing categorical cross entropy loss and center loss [46] using mini-batch stochastic gradient descent (SGD) with momentum. We also apply dropout regularization [47] before feeding $\mathbf{f}$ into the classifier to prevent co-adaptation of hidden units.

IV. Experiment

A. Datasets

We use three commonly studied sentiment analysis datasets to evaluate our model, including Yelp Review Polarity (Yelp P.), Yelp Review Full (Yelp F.) and IMDB Movie Review. These datasets are diverse in the aspect of number of sentiment classes, average text length, and number of samples. The statistics of the three datasets are shown in Table II.

TABLE II: Statistics of the three sentiment analysis datasets used in our experiments.

Dataset	**# of classes**	**Average length**	**Train samples**	**Test samples**
Yelp P.	2	156	560k	38k
Yelp F.	5	158	650k	50k
IMDB	2	292	25k	25k

Yelp P. and Yelp F. datasets are constructed from Yelp Dataset Challenge 2015 by Zhang et al. [10], which contain crowd-sourced reviews about businesses. Yelp P. is a binary sentiment classification dataset whose sentiment polarity is either positive or negative; Yelp F. contains more fine-grained sentiment classes ranging from rating 1 to rating 5. IMDB dataset is constructed from IMDB movie reviews by Maas et al. [48]. It is a binary sentiment classification dataset with even number of positive and negative reviews. Compared with Yelp reviews, IMDB movie reviews are much more longer, normally in document-level.

B. Experiment Settings

1) Baseline models: To study the performance of our proposed model, the following models are used as baseline models for comparison:

CNN-based models including conventional Word-level CNN and Char-level CNN [10], as well as a very deep CNN namely VDCNN for text classification [24].

RNN-based models including standard LSTM, discriminative LSTM (D-LSTM) implemented by [13] and Skim-LSTM which updates the hidden states dynamically [49].

Attentive models including label-embedding attentive model (LEAM) [50], Bi-directional LSTM with self-attention (BiLSTM+Self-attention) for sentence embedding representation [34] and Transformer encoder [15] for text classification.

For our proposed ConvTransformer, we also study its performance without external knowledge, i.e. without sentiment embedding in the sentiment-aware attention mechanism.

2) Hyper-parameters settings: We set aside 10% of training data as development set to tune model hyper-parameters and adjust learning rates. The evaluation metric is classification accuracy and we report the average accuracy on test set based on 5 independent trainings of the model.

The word embedding matrix $\mathbf{W}^{wrd}$ is constructed using 300-dimensional Glove word embeddings [51] and it is fixed in the model. The fully connected layer in the classifier has a dimensionality of 100. Dropout rate is set to 0.4. ReLU is used for all nonlinear activation functions except sentiment-aware attention which uses tanh. The attention dimension in sentiment-aware attention is set to 200, and the dimensions of sentiment embedding and position embedding are $d_s = 20$ and $d_p = 60$ respectively. For our proposed ConvTransformer, we use 3 layers with 6 convolutional heads in each layer and $m = 100$ convolutional filters are used in each head. We use three convolution window sizes $n = 2, 3, 4$ in ConvTransformer, each is applied in two heads. For Transformer encoder, we use

TABLE III: Classification accuracy (%) on three sentiment analysis datasets. * means the results are obtained from our implementation. – means not reported. All other results are directly cited from the respective papers.

Model	Yelp P.	Yelp F.	IMDB
Word-level CNN [10]	95.40	59.84	89.08*
Char-level CNN [10]	94.75	61.6	–
VDCNN [24]	95.72	64.72	–
LSTM	95.37*	64.01*	89.14*
D-LSTM [13]	92.60	59.60	–
Skim-LSTM [49]	–	–	91.20
LEAM [50]	95.31	64.09	–
BiLSTM+Self-attention [34]	96.08*	65.79*	89.56*
Transformer Encoder [15]	96.13*	65.34*	88.05*
ConvTransformer (w/o sentiment embed)	97.14	67.60	92.00
ConvTransformer (with sentiment embed)	**97.46**	**68.39**	**92.91**

the default implementation as in [15] with the same number of heads and layers as ConvTransformer.

For training, the initial learning rate is 0.01 with a momentum of 0.9, and it is decayed with a rate of 0.9 after 10 epochs if the classification accuracy on development set does not improve. The learning rate and weight for center loss are 0.1 and 0.001 respectively. Batch size is set to 100 and we train the model for 70 epochs.

C. Results and Analysis

Table III shows the test accuracy of our model as well as baseline models on the three sentiment analysis datasets. Results show that our model achieves the best performance on all the three datasets, outperforming CNN-based, RNN-based and attentive baseline models.

For CNN-based models, results show that Character-level CNN is more effective for fine-grained sentiment analysis than binary classification, and deep CNN demonstrates performance improvement over shallow CNNs. Our proposed ConvTransformer (without sentiment embedding) shows significant improvement over conventional CNNs including deep CNN. This is due to the benefits of the multi-head structure with multi-window convolution that allows our model to capture more semantic and syntactic features from different n-grams, as well as the sequential output with attention mechanism that preserves important position information for final representation.

RNN-based models are very sensitive to parameter settings. Our implementation of LSTM performs better than D-LSTM. Skim-LSTM shows good performance on document-level sentiment analysis (IMDB dataset), probability due to its special treatment of unimportant inputs that alleviates gradient vanishing problem. The proposed ConvTransformer uses convolution operation that does not suffer from gradient vanishing problem.

Attentive models especially self-attention-based models outperform RNN-based models significantly for short texts (Yelp P. and Yelp F.). However, the performance degrades for long texts (IMDB). This is because that the model is affected by irrelevant words and not able to focus on important keywords since self-attention mechanism focuses on global context. Our proposed ConvTransformer focuses more on local context using n-gram convolution and is able to capture important keywords and phrases more effectively.

TABLE IV: Ablation study on ConvTransformer. Classification accuracy is reported on the development set of IMDB.

Model	Dev Acc (%)
ConvTransformer	93.11
1. − Multi-head	92.05
2. − Multi-layer	91.74
3. − Multiple window size	92.32
4. − Sentiment embed.	92.07
5. − Position embed.	91.94

Results also show that incorporating external knowledge of words' sentiment in sentiment-aware attention mechanism further improves the performance of our model on all the datasets. This demonstrates the benefits of utilizing external knowledge as well as the effectiveness of the proposed sentiment-aware attention mechanism.

D. Discussions

1) Ablation Study: To study the contributions of specific parts of our model, we perform ablation study on the development set of IMDB by comparing the performance of our model with and without the specific part. Experiment results are shown in Table IV.

(1) We remove the multi-head structure by using only one head in each ConvTransformer layer. Result shows that the performance degrades by 1.14%, demonstrating the effectiveness of jointly performing n-gram convolutions in different sub-word spaces. (2) We remove the multi-layer structure by using single ConvTransformer layer. The performance degrades of 1.47% demonstrating the benefits of using multi-layer ConvTransformer for text encoding. The upper layer of ConvTransformer is able to widen the convolution context and induce more abstract and discriminative representations of texts. (3) In all the convolutional heads, we use single convolution window size of 3 instead of multiple window sizes, the performance drops by 0.85%. This proves the effectiveness of using different convolutional heads to focus on n-grams with different lengths. (4) After removing the sentiment embedding in sentiment-aware attention, the performance drop by 1.23%. This demonstrates the benefits of incorporating words' sentiment knowledge from external knowledge base into the model using the proposed sentiment embedding construction and sentiment-aware attention methods. (5) After removing the position embedding in sentiment-aware attention, the performance drop by 1.25%. This demonstrates the importance of preserving and capturing the positional or sequential information of texts.

2) Attention Visualization: To study what does our model focus on, we conduct attention weights visualization for sentiment-aware attention. Table V shows the visualization results of three sample sentences from the test set of IMDB.

It is observed that our model pays more attention to words and phrases with high sentiment intensity, either positive sentiment such as"delightful", "entertaining" and "Recommended" in the first sample sentence, or negative sentiment such as

TABLE V: Visualization of attention weights for sentiment-aware attention. Words are highlighted based on the attention weights assigned to them. Best viewed in color.

#	Sample Sentences	Sentiment
1	Make sure you make this delightful comedy part of your holiday season ! If you admire Dennis Morgan or Barbara Stanwyck , this film is a fun one to watch . They really work well together as you would see in this movie . The whole cast was very entertaining . Since I 'm a Dennis Morgan fan , this film was a real treat ! But ... everyone can enjoy it ! Recommended !	positive
2	This movie really has nothing going for it . With the Reverend played by Phillip Seymour Hoffman complaining about his constipation and other toilet humor in a 2.5 hour movie , you know that they made no cuts at all and left the crap in , literally . It 's a waste of good talent , and a total embarrassment . Dreadful !	negative
3	In this Silly Symphony , a mouse from the country visits his cousin in the city . Most of the short is the two mice exploring the dinner table . The animation is fine , where this short suffers is in a lack of humor . Perhaps I 've just seen this `` dinner table adventure " in one too many Tom and Jerry shorts . Even though this came first , I just did n't find it that enjoyable .	negative

TABLE VI: Comparison of model parameters and average inference time per batch (batch size of 100) on IMDB dataset.

Model	Trainable param. #	Inf. time per batch
Transformer	3.38 Million	184 ms
ConvTransformer	1.50 Million	62 ms

"crap", "waste" and "embarrassment" in the second sample sentence. Besides, our model is able to capture the negation of a sentiment word effectively. For example, the negation of word "enjoyable" in the third sample sentence. Hence, the overall sentiment polarity of the text can be capture correctly.

3) Computational Efficiency: To evaluate the computational efficiency of our model, we compare the number of trainable parameters and inference time of our model with Transformer. The evaluation is conducted on IMDB dataset with batch size of 100 using NVIDIA Tesla P40 GPU. Results are shown in Table VI. Compared with Transformer, our proposed ConvTransformer has 56% less trainable parameters and 3 times faster inference speed. Therefore, ConvTransformer is a more light-weight and efficient model for sentiment analysis.

V. Conclusion and Future Work

We propose a novel neural architecture called ConvTransformer that combines the advantages of CNN and Transformer for text representation. ConvTransformer has a multi-head structure that jointly performs different n-gram convolutions in different sub-spaces. It is more effective in capturing local n-gram features compared with CNN and Transformer and also able to preserve the sequential information of texts. Moreover, we incorporate external knowledge of words' sentiment intensity into our model by utilizing an external knowledge base, SentiWordNet. We propose a sentiment-aware attention mechanism that takes each word's sentiment intensity and position information into consideration while obtaining the final text representation. Extensive experiments and analyses show that our model is effective for sentiment analysis, outperforming conventional CNN, RNN and attention-based models.

In future work, we will apply our proposed ConvTransformer in other text classification tasks and incorporate other types of knowledge such as conceptual knowledge into sentiment-aware attention mechanism. Besides, we will explore the potential applications of ConvTransformer on sequence-to-sequence NLP tasks such as machine translation.

References

[1] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 417–424.

[2] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2nd international conference on Knowledge capture*. ACM, 2003, pp. 70–77.

[3] Y. Lu, X. Kong, X. Quan, W. Liu, and Y. Xu, "Exploring the sentiment strength of user reviews," in *International Conference on Web-Age Information Management*. Springer, 2010, pp. 471–482.

[4] M. Eirinaki, S. Pisal, and J. Singh, "Feature-based opinion mining and ranking," *Journal of Computer and System Sciences*, vol. 78, no. 4, pp. 1175–1184, 2012.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[6] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 412–418.

[7] M. R. Saleh, M. T. Martín-Valdivia, A. Montejo-Ráez, and L. Ureña-López, "Experiments with svm to classify opinions in different domains," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14 799–14 804, 2011.

[8] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," *Decision support systems*, vol. 57, pp. 77–93, 2014.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[10] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[11] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69–78.

[12] J. Nowak, A. Taspinar, and R. Scherer, "Lstm recurrent neural networks for short text and sentiment classification," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 553–562.

[13] D. Yogatama, C. Dyer, W. Ling, and P. Blunsom, "Generative and discriminative text classification with recurrent neural networks," in *Thirty-fourth International Conference on Machine Learning (ICML 2017)*. International Machine Learning Society, 2017.

[14] M. Seo, S. Min, A. Farhadi, and H. Hajishirzi, "Neural speed reading via skim-rnn," *ICLR*, 2018.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[16] X. Cheng, W. Xu, T. Wang, W. Chu, W. Huang, K. Chen, and J. Hu, "Variational semi-supervised aspect-term sentiment analysis via transformer," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 2019, pp. 961–969.

[17] M. Jiang, J. Wu, X. Shi, and M. Zhang, "Transformer based memory network for sentiment analysis of web comments," *IEEE Access*, vol. 7, pp. 179 942–179 953, 2019.

[18] X. Li, L. Bing, W. Zhang, and W. Lam, "Exploiting bert for end-to-end aspect-based sentiment analysis," in *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 2019, pp. 34–41.

[19] P. Li, K. Mao, X. Yang, and Q. Li, "Improving relation extraction with knowledge-attention," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 229–239.

[20] P. Zhong, D. Wang, and C. Miao, "Knowledge-enriched transformer for emotion detection in textual conversations," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 165–176.

[21] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 655–665.

[24] A. Conneau, H. Schwenk, Y. LeCun, and L. Barrault, "Very deep convolutional networks for text classification," in *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017*. Association for Computational Linguistics (ACL), 2017, pp. 1107–1116.

[25] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 562–570.

[26] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[27] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional lstm with two-dimensional max pooling," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 3485–3495.

[28] B. Wang, "Disconnected recurrent neural networks for text categorization," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2311–2320.

[29] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[30] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional cnn-lstm model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 225–230.

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[32] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[33] Y. Wang, M. Huang, L. Zhao *et al.*, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.

[34] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[35] Z. Zhang, Y. Zou, and C. Gan, "Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression," *Neurocomputing*, vol. 275, pp. 1407–1415, 2018.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[37] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[38] P. Li and K. Mao, "Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts," *Expert Systems with Applications*, vol. 115, pp. 512–523, 2019.

[39] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowledge-Based Systems*, vol. 163, pp. 955–971, 2019.

[40] F. H. Khan, U. Qamar, and S. Bashir, "A semi-supervised approach to sentiment analysis using revised sentiment strength based on sentiwordnet," *Knowledge and information Systems*, vol. 51, no. 3, pp. 851–872, 2017.

[41] Y. Wang and P. Li, "Knowledge-oriented hierarchical neural network for sentiment classification," in *IOP Conference Series: Materials Science and Engineering*, vol. 646, no. 1. IOP Publishing, 2019, p. 012023.

[42] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining." in *LREC*, vol. 6. Citeseer, 2006, pp. 417–422.

[44] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." in *Lrec*, vol. 10, no. 2010, 2010, pp. 2200–2204.

[45] C. Fellbaum, "Wordnet," in *Theory and applications of ontology: computer applications*. Springer, 2010, pp. 231–243.

[46] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[48] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 2011, pp. 142–150.

[49] M. Seo, S. Min, A. Farhadi, and H. Hajishirzi, "Neural speed reading via skim-rnn," *International Conference on Learning Representations*, 2018.

[50] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2321–2331.

[51] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.