This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

Spatial Pyramid Pooling Mechanism in 3D Convolutional Network for Sentence-Level Classification

Xi Ouyang, Kang Gu, and Pan Zhou*, *Member, IEEE*

***Abstract*—In this paper, we investigate the usage of the convolutional neural network (CNN) to propose a novel end-to-end language processing structure to model textual data for this task. In particular, we propose a 3D CNN structure for the task, which is featured by Spatial Pyramid Pooling (SPP). To our knowledge, it is the first time that 3D convolution and SPP structure are applied together in language processing issues. Compared with methods of 2D CNNs, the proposed method can effectively and efficiently capture the complicated internal relations in sentences. Furthermore, in previous work, the issue of sentence length variety is usually addressed by padding zero to make all sentences vectors to a fixed length, which causes too much redundant and useless noise. Inspired by the SPP structure for object detection in image processing, this issue can be well handled with the Spatial Pyramid Pooling, which divides the sentences into several length sections for respective pooling processing. Experiments are conducted for the task of sentence classification as well as relation classification. Experiments on Stanford Treebank, TREC, subj and Yelp datasets demonstrate that our proposed method can outperform other state-of-the-art models, with respect to classification accuracy. Auxiliary attempts to leverage our method to SemEval-2010 Task 8 dataset further substantiate the model's capability of extracting features efficiently.**

***Index Terms*—3D CNN, SPP, Sentence Classification, Relation Classification.**

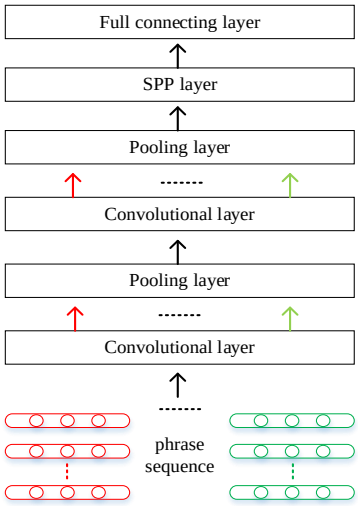


Fig. 1. This figure displays the abstraction of the proposed network. Original sentences are separated into several length sections, and then are processed to form according phrase sequence. In this case the phrase sequence is composed of 3-word-phrase, whose unique color denotes according length after padding zero and attuned pooling process. More detailed fragments will be discussed in section IV.

I. Introduction

CLASSIFICATION tasks in NLP field have been extensively explored, such as sentence classification and relation classification. The former task aims to categorize sentences into different classes according to their intrinsic attributes, such as sentiment, subject and emotion. It is becoming an essential research topic in the field of natural language processing due to its wide applications in real life. While the demand for this task is growing, current research in this area is still immature. Developing an accurate classifier which can effectively extract complicated correlations in sentences remains a challenging endeavor.

Recent remarkable performance in computer vision and achieved by deep learning inspires researchers to exploit this new tool to address sentence classification problem. Such deep learning-based practices can be categorized into two types, namely Convolutional Neural Networks (CNNs) based models, and Recurrent Neural Networks (RNNs) based models. CNN-based models employ locally connected filters to capture word dependency in sentences [1]. Although these methods are fast in terms of training and inference, the local scanning of convolution operation is usually not adaptive to complicated sentence structures, as it often overlooks important information between words. On the other hand, the structure of RNN is consistent with semantic structures in sentences, making them a better structure for sentence modeling [2]. However, RNN-based methods suffer from high computational complexity, which makes them require substantially longer training and inference time compared to CNN.

There exist several attempts which aim to improve the CNN's capability of capturing sequential and hierarchical correlations in complicated natural language sentences structures. A quite effective method among them is to add extra explicit or implicit phrase-level features to raw input sentences [3], [4]. The intuition is that correlations between words will gradually fade away with the increase of distances between them, such that adding these features can summarize important local correlation which can help CNN for better understanding. There-

Xi Ouyang and Kang Gu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. Email: ouyang@hust.edu.cn, kanggu@yahoo.com.

*Corresponding author: Pan Zhou is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. Email: panzhou@hust.edu.cn.

This work was supported in part by National Science Foundation of China with Grants 61401169.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

fore in our design, we substitute input sentences with n-gram sequences as hand-crafted features which extends the raw input sentence embeddings to a further dimension. Specifically, the new dimension is introduced by splitting original sentences are into phrases of k words. We subsequently concatenate the processed of k-word phrases in a further dimension and pass them as input to the neural network. This enforces the model to exam every n-gram to capture information as intact as possible by scanning these per-defined features. We further employ 3D convolutional layers [5] to extract neighboring correlations, as 3D CNN can preserve important information in every dimension contained in phrases.

In addition, lengths of sentences have significant disparity, while traditional model can only accept fixed-length sentences as input. In order to tailor all sentences to models, a common practice is to pad all sentences with zeros to a maximum length. However, short sentences may suffer from adverse impacts brought out by the padding operation. This will introduce serious input noise, eventually affecting the model performance. In order to solve this problem, we incorporate Spatial Pyramid Pooling (SPP) [6] into our architecture, so that padding requirement for sentences is reduced. We extend the original SPP layer into 3D phrase pooling to weaken the boundary effect brought by great length variability. This operation allows CNNs to accept flexible input size, which is one of the key advantages of RNNs [7], [8], [2]).

The system-level visualization of our model is displayed in Fig.1. The n-gram sequences encode rich local dependencies so that that we can enforce the model to exam every n-gram to capture information as entirely as possible, rather than let itself produce n-gram features.

We summarize contributions of this paper as follows:

- We substitute original sentences with n-gram sequences to extend the input dimension, and subsequently apply 3D convolutional layers to capture local correlations between n-gram for two tasks. This allows model to learn more complicated word-wise dependency compared to 2D CNN.
- We upgrade the original SPP layer into a 3D phrase pooling operation, to enable CNN architectures to accept sentences with various lengths without massive zero padding effort. This significant reduces noise and eventually improves the model performance.
- Experiments conducted on 4 benchmark datasets for sentence classification and one benchmark dataset for relation classification demonstrate that our proposed architecture achieves competitive results, as compared with the other state-of-the-art models.

To the best of our knowledge, it is the first time that 3D convolutional layers and SPP layers are applied simultaneously to solve NLP problems. We believe it is promising to extend this architecture to address other sentence-level problems. The rest of this paper is organized as follows. Section II presents related work of sentence classification. Meanwhile, we discuss the background of the proposed model in Section III. Section IV describes the structure of deep model and the important components in detail, respectively. Our experimental results for sentence classification are presented and analyzed in Section V. Moreover, additional attempts to leverage our model to the task of relation classification are included in section VI. Finally, we conclude the study in Section VII.

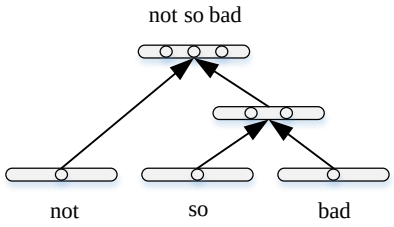


Fig. 2. This figure shows how vector representation of phrase is computed recursively in RvNN.

II. Related work

A. Traditional Approaches

NLP is a theory-motivated range of computational techniques, designed for the automatic analysis and the representation of human language [9]. Related tasks include machine translation [10], [11], [12], word prediction [13], [14], question answering [15] and so on, among which sentence classification plays a fundamental role to explore semantic features of textual data. Its purpose is to determine the sentiment polarity (positive or negative) of a sentence (or a document) based on its textural content. Hitherto, this task has already attracted extensive interests from both research and industry communities.

We will introduce the traditional approaches briefly and focus our attention on deep learning models. For the traditional approaches We mainly describe two dominated directions for sentence-level sentiment classification, namely lexicon-based approach and corpus-based approach.

Lexicon-based methods [16], [17], [18] typically utilized existing sentiment lexicons of words and phrases, each of which was annotated with its sentiment polarity or sentiment length. Usually, linguistic rules were leveraged to promote the prediction of sentiment for sentences (or documents).

Corpus-based approaches treated sentiment classification as a special case of text categorization problem [19]. They mostly relied on sentences with annotated sentiment-polarity to build sentiment classifier. Socher *et al.* [36] reported the prominent performances of using Naive Bayes (NB), Support Vector Machines (SVMs) as well as Naive Bayes with bag of bigram features. In their experiments, SVM based on bag of words features achieved the best accuracy among three mentioned models. Wang *et al.*'s work [20] combined generative and discriminative classifiers, presenting a simple model variant (NBSVM) where SVM was built over NB log-count ratios as feature values. NBSVM has also been proven to be a strong and robust performer over all the tasks included in the work. While they also pointed out that the performances of SVMs vary drastically depending on the model variant, features used, task and dataset.

B. Deep Learning

With the prosperity of deep learning, many recent studies began to leverage deep neural networks as feature extractors

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

to learn discriminative representation over a piece of documents. Recursive neural networks (RvNN), which utilizes the recursive structures of inputs to obtain their global representations, achieve success in many NLP tasks including sentence classification. Socher *et al.* [36], [21] demonstrated that the recursive tree structures of RvNN could let information from leaf nodes and its internal nodes get combined in a bottom-up manner through the tree. Another network (DRNN) [22] constructed by stacking multiple recursive layers on top of each other succeeded in modeling sentences according to the tests on Stanford Sentiment Treebank [36]. Fig. 2 displays the way of RvNN to obtain the polarity of the phrase *"not so bad"* is computing the words *"so"* and *"bad"* firstly, then combining *"not"* and *"so bad"* [23].

Recurrent Neural Networks (RNNs) are also very prevailing models in NLP, since their recurrent structure is very suitable to process variable-length texts. Especially, Long Short-Term Memory (LSTM) has been proved be an efficient RNN-based structure for NLP tasks, which is a special kind of RNN and capable of learning long-term dependencies. Recursive Neural Tensor Network (RNTN) [36] was designed to be able to produce hidden vector of parent node via combining two descendant nodes, with tensor to capture second-degree polynomial interactions. Tai *et al.* [24] and Zhu *et al.* [7] both generalized LSTM to Tree-LSTM, where each LSTM unit gained information from its children units. Tree-LSTM indeed can be viewed as considering together a recursive neural network and a recurrent neural network. Besides, two levels of attention mechanisms, word attention and sentence attention, were incorporated into a hierarchical network for document classification [25], producing excellent results. Zhou *et al.* [26] introduced bidirectional-LSTM (BLSTM) with attention mechanism to automatically select features, which have a decisive effect on classification.

Convolutional Neural Networks (CNNs) typically function in this way: convolving filters are applied to local features to perform feature mapping and then pooling operation is used over the time-step dimension to obtain a fixed-length output. Most existing works can be categorized into two subsets: providing more effective input for CNN [27], [28], [29], [30] or designing novel pooling strategies [1], [31], [32], [33] to reduce damage to global information. Kim [45] presented a multi-channel CNN with two sets of word vectors, static vectors from `word2vec` [34] and fine-tuned vectors for the task. Experimental results proved that learning task-specific vectors through fine-tuning led to further improvement in accuracy. Multi-Group Norm Constrained CNN (MGNC-CNN) [35], which capitalized on multiple sets of word embeddings for sentence classification, adopted a group regularization strategy of differentially penalizing weights associated with the subcomponents generated from the respective embedding sets. Zhang *et al.* [40] offered an empirical exploration of one-hot-CNN, which used character-level features and required no knowledge of words. Above methods managed to evaluate the effectiveness of more diversified input for CNNs, making us believe CNNs' potentials of extracting features. Meanwhile, powerful pooling strategies are also expected to make a difference to the final outcomes. Kalchbrenner *et al.* [1] put

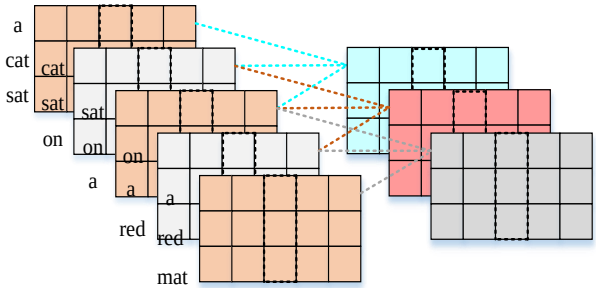


Fig. 3. The picture displays the convolution operation when phrase is of size 3. Those phrases are produced through a window of the same size sliding along the sentences and each row is a 300-dimensional word vector. Note that the input should be $5\times3\times300$ which means the sentence has five 3-word phrases. Our 3D phrase convolution has 3-dimensional operation corresponding for this input. For instance, if using a convolutional kernel of $3\times1\times1$, we can obtain the output of $3\times3\times300$ as shown in this figure.

forward dynamic-k-pooling whose pooling rate was computed from a linear function of input length. A recent two-way attentive pooling mechanism [31] could learn how to compute interactions between the items in the input pair, while it is extensible for both CNNs and RNNs.

By incorporating both CNN and RNN, recurrent convolutional neural networks [41], [42], [43], [44] are drawing more and more attention among researchers gradually. RCNNs typically apply a recurrent structure to capture contextual information to the greatest extent, then employ a pooling layer (usually a max-pooling layer) which selects those key features for classification. Compared to those works, our model can capture more local word-level correlation and handle variable-size sentences.

The biggest challenge for sentence classification is to create a more efficient structure to capture the correlations between words in sentences. In previous works, researchers typically applied CNN or RNN models to explore more distinctive representation for the sentences. However, we find local interdependences between nearby words are more important in sentences. To strengthen the capability of capitalizing on local interdependences, we introduce a novel 3D CNN structure for this task. Meanwhile, it would bring about too much redundant noise when padding all sentences zero to a maximum length. Inspired by the Spatial Pyramid Pooling structure in object detection [6], we also propose novel phrase-level spatial pyramid pooling layers to solve this issue (Fig.1).

III. Our Architecture

We begin with illustrating the entire framework in detail and further introduce important technologies used in our network, 3D Phrase Convolution and 3D Phrase Pooling.

Framework We propose a deep neural network tailored to sentence classification by integrating 3D convolution and pooling layers. We illustrate the architecture in Figure 4. This architecture consists of two 3D phrase-level convolutional layers (C1 and C2), two 3D phrase pooling layers (P1 and P2), a fully-connected layer (F1) and a softmax layer in the end. We apply the max pooling in both two 3D phrase pooling layers. Usually, to input sentences into neural networks, the words in each sentence will be encoded to word vectors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

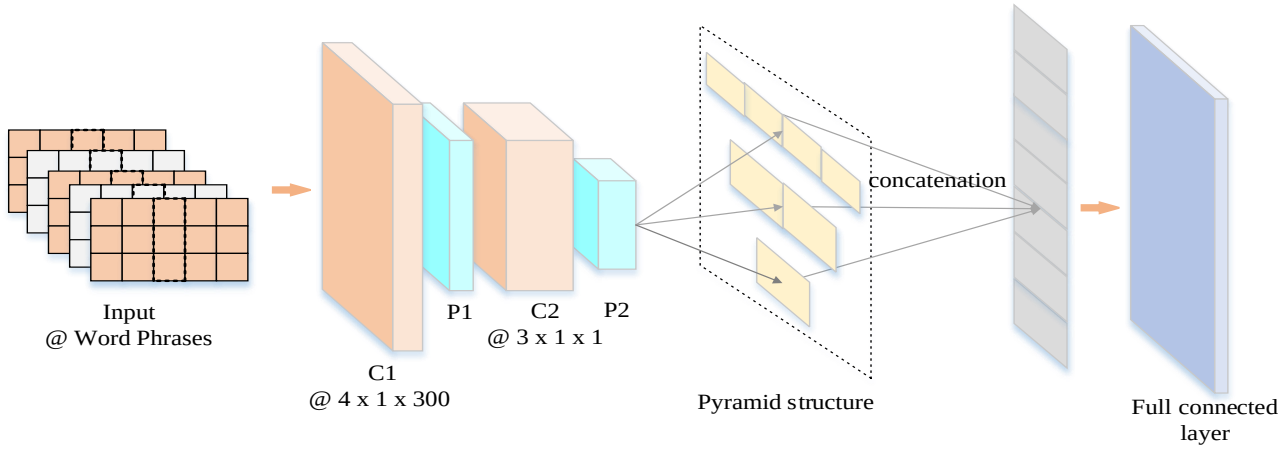


Fig. 4. A 3D-SPP architecture for sentence classification is shown as above. This network contains two 3D convolution layers, two 3D SPP pooling layers and a full connection layer. Details are illustrated in the text. Note that this network only demonstrates the one case of 3-word-phrase input but it still suffices to represent the complete architecture. Orange boxes refer to our 3D CNN layers for word phrases and blue boxes refer to our spatial phrase pooling layers.

`Word2vec` [34] and One-hot encoding are applied widely for this purpose. Through this operation, we encode each sentence into a 2D "image" and each word forms a vector with 300 elements. In order to capture local word-level correlation in sentences, we split original sentences into overlapping phrases of k words and each k-words phrases is $k \times 300$ matrix. Then the whole sentence consists of several such matrices, and each phrase matrix can be regarded as a frame in "video". A 7-word sentence can transformed into $5 \times 3 \times 300$ "video" data for 3D CNNs if we split original sentences into overlapping phrases of 3 words. We illustrate this operation in Figure 3. Denoting the sequential length of input overlapping phrases in each original sentence as l, an input with size of $l \times 1 \times k \times 300$ will be firstly processed by C1 (1 represents for the channel of input and k represents for the length of phrase). The kernel size of C1 is set to $4 \times 1 \times 300$ thus the output of C1 is of $(l-3) \times n_1 \times k \times 1$ (n_1 is the number of filters in C1). Let $l' \times n_1 \times k' \times 1$ denotes the size of the output of P1 pooling layer. Subsequently, the output of P1 would enters C2, whose kernel size is of $3 \times 1 \times 1$. As we set the number of filters in C2 layer to n_2, the convolution operation will produce output of $(l'-2) \times n_2 \times k' \times 1$. Therefore, the output of P2 is expected to be $4 \times n_2 \times 1 \times 1$. We subsequently employ a 3D phrase pooling layer obtain a fixed length of vector length after P2. Specific operations of these operations will be detailed next.

Before entering the fully-connected layer (F1), we use a pyramid structure for feature concatenation, as we show in Figure 6. To form a pyramid structure, a group of pooling kernels with strides of 1, 2, 4 respectively is performed on the output of P2 respectively (only in the sequential dimension). Thus we obtain three parts of feature maps: 4, 2 and 1 in the first channel, which are later concatenated to form the input of fully-connected layer. Therefore feature maps with size of $7 \times 1 \times 1$ are sent into F1. The SPP mechanism is built up by the two 3D phrase pooling layers and the pyramid structure. After that, the final prediction is made by a softmax layer.

Defining 3D Phrase Convolution 3D convolutional neural network [5] was originally designed for action recognition, as it can extract features from both spatial and temporal dimensions by performing 3D convolutions. It captures the motion information encoded in multiple adjacent frames. We improve the original CNN in this task to 3D CNN to deal with n-gram input. Since traditional 1D or 2D CNNs (the idea of one-dimensional convolution is to take dot product between convolutional kernel and n-gram in the sentence [1]) treat sentence as a concatenation of words, the important relations within several adjacent words might be impaired by trivial pooling operations. Our model strengthens the capability of capturing local word-level correlation, mainly contained in the phrases, while still keeping the initial order of the words well.

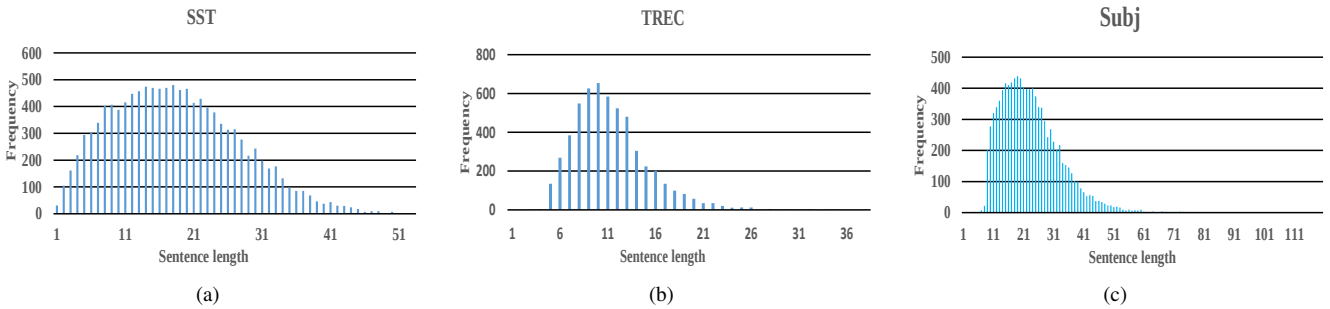
Let $X_{1:N}$ denote the sequence with length N. Each $x_n \in X_{1:N}$ is a d ($d = 300$ in our experiments) dimensional vector $x_n \in R^d$ to encode the n-th word. In our design, we split original sentences into overlapping phrases of k words and each k-words phrases is $k \times 300$ matrix. Then the whole sentence consists of several such matrices. We show an example is illustrated in Figure 3. We use y_n^k, a $k \times d$ matrix to represent a k-word phrase $x_{n:n+k}$, where k is the dimension of the length of phrases and d is the dimension of word embedding vectors. $x_{n:n+k}$ denotes a concatenated collection of $x_n, x_{n+1} \dots x_{n+k}$. We can then define the input for our 3D convolutional network as:

$$y_{n:n+l}^k = x_{n:n+k} \oplus x_{n+1:n+k+1} \oplus \dots \oplus x_{n+l:n+k+l}.$$

Note that $y_{n:n+l}^k$ is a $l \times k \times d$ matrix and $\oplus$ is a concatenation operator. Through this operation, our input becomes a 3-dimensional matrix ($l \times k \times d$). Our phrase-level CNN is built up by 3D convolutional kernel $W \in R^{p,q,r}$, where (p, q, r) is kernel size. The (p, q, r) means the convolution operation sizes for the different dimensions of phrases, the length of each phrase and the word vector. Then the feature maps feeding into the convolutional layer are connected to form multiple contiguous phrase-level features in the previous step, thereby capturing more local word-level correlation. Formally, the jth feature map in ith layer is defined as:

$$V_j^i = f(\sum_m W_{ijm}^{p_i q_i r_i} \times V_m^{i-1} + B).$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing



(a) (b) (c)

Fig. 5. The statistics of sentence length of our datasets. (a) We collect the statistics of frequency of every existing length in Stanford Sentiment Treebank. The frequency is distributed on a wide range and obviously centered in the section of $[1, 30]$. (b) The statistics of sentence length of TREC. The distribution of frequency is mainly limited to the section of $[5, 20]$, a quite narrow range compared to that of SST. (c) The statistics of sentence length of Subj. The distribution of frequency is mainly limited to the section of $[10, 40]$.

Here, B is a bias term which is accordingly three-dimensional bias and f is a non-linear function such as the hyperbolic tangent. V_m^{i-1} is the mth feature map in the previous layer $(i-1)$. When $i = 1$, V^0 means the 3-dimensional input $y_{n:n+l}^k$.

Defining 3D Phrase Pooling Spatial Pyramid Pooling (SPP) [6] can generate a fixed-length representation regardless of input scales in the task of visual recognition. The inspiration of SPP is introduced into our architecture as a good solution to length variety trouble. However, there are two difficulties for applying it from images to our research. First, original SPP expects the 2D input as such images, but our input is of 3-dimensional. Therefore, we adopt it to 3D phrase pooling by add an addition dimension in our pooling kernels. Second, the network with SPP expects multi-size image for training [6]. However, the sentence vector matrices as the input of CNNs, can neither be compressed nor cropped easily since vital messages might get destroyed. Therefore we divide sentences into different sections according to their actual length and padding them zero to the upper boundary of the each section. Specifically, considering the datasets (Stanford Treebank [36], TREC [37], Subj [38], and SemEval-2010 Task 8 dataset [52]) we used, We divide sentences of those dataset into four sections, and the lengths of each are $[17, 24, 44, 53]$. As sentences are already rationed to respective section, they are padded with zero to the upper boundary of the section. Through this operation, short sentences will not be padded with massive zeros which avoids serious zero noise pollution. Meanwhile, this allows our model only need to consider several predefined input lengths instead of arbitrary lengths.

Considering we only have several certain predefined different lengths, we expect that we can gain the fixed length of vectors from P2 layer as shown in Fig. 4. This can include the original local interdependences in sentences as much as possible. Let the i_1 and i_2 denote input vector lengths for those two layers and o_1 and o_2 denote output vector lengths, then the pooling ratio p_1 and p_2 in P1 and P2 layer can be computed as:

$$p_1 = \frac{i_1}{o_1}, p_2 = \frac{i_2}{o_2},$$

To obtain the fixed length m output of P2 layer, given the input length $l \in L$ $(m \leq l)$, the setting of P1 and P2 can be expressed as:

$$\lceil \frac{l}{m} \rceil = p_1 \times p_2.$$

Here $\lceil \frac{l}{m} \rceil$ is the operation of rounding outcome up to an integer, which represent the ratio of the the input length l and the feature length m after P2 layer. In our design, our pooling layers have 3-dimensional kernel of size $[n_1, n_2, n_3]$ corresponding to the input size $l \times k \times d$, where l is the length of input overlapping phrases, k represents for the length of phrases, and d is the word vector dimension. For all the input sentences of Stanford Sentiment Treebank, the length of phrases and the word vector dimension are fixed while the the lengths l of input overlapping phrases is different because the different sentence length sections ($[17, 24, 44, 53]$). Therefore, we only need to decide the pooling kernel size n_1 corresponding to this dimension. Meanwhile, we set the other two pooling kernel sizes (n_2, n_3) to 1 for the length k of phrases and the word vector dimension d. Then we split original sentences into overlapping phrases of k words, resulting in length l to be $[18 - k, 25 - k, 45 - k, 54 - k]$. To obtain the fixed length output of P2 layer, we need to consider the two sets of pooling parameters (let (p^1, p^2, p^3, p^4) denote the pooling kernel size n_1 of first pooling layer P1 and (q^1, q^2, q^3, q^4) for the second one P2. both p and q are positive integers). Here our choice is $(1, 2, 3, 3)$ and $(3, 2, 4, 5)$. Through all those methods, we can get a fixed-length output of P2 layer for all different sentences.

For several outlines in our dataset whose length is beyond 53 and can not be divided into our any section $[17, 24, 44, 53]$, the first 53 words can be selected into our model. Actually, new pooling kernel sizes n_1 for P1 and P2 can be set to handle the length beyond 53. In this paper, we find the lengths of most sentences will not beyond 53 as we show the statistics of sentence length in Fig. 5. Alos, for the SemEval-2010 Task 8 dataset, there are only 25 sentence that has more than 53 words. Most sentences are location the section between 5 and 40, so we only set four length sections in our paper.

A. Training objective

As the same in [36], we apply *softmax* layer to classify the predictions for classes. We aim to minimize the cross-entropy error for each neuron during training. It is computed by regularized sum (the first term is the cross-entropy error over all the neurons, and the second term is a L_2-regularization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

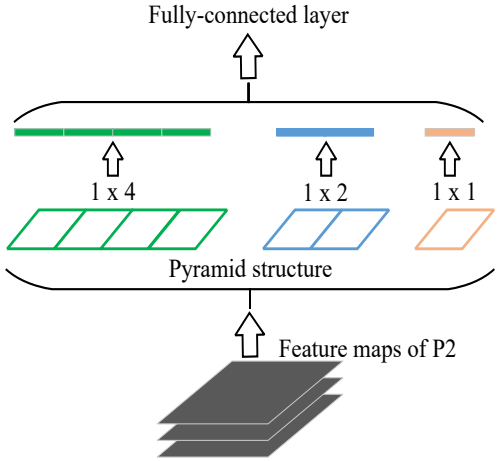


Fig. 6. The final operation of SPP is shown above. Note that it is only a special case to produce 4 , 2 and 1 pyramid and parameters should better be adjusted to practice. The point of this action is actually to increase the proportion of active features.

TABLE I
HYPERPARAMETERS

Parameter	Parameter Name	Value
d	Word Embedding Size	300
p	Word Position Embedding Size	25
λ	Coefficient of L_2-regularization	0.001
η	Learning rate	0.01
k	Phrase length	chosen from $[2, 3, 4, 5]$
$Epoch$	Train Epoch	200
$Batch$	Batch Size	64

penalty) :

$$E(\theta) = -\sum_i \sum_j p_j^i log(y_j^i) + \lambda ||\theta||^2, \quad (1)$$

where y_j^i is predicted distribution and p^i is the target distribution. Besides, p^i is a binary vector. If the desired output class is h, we set p_h^i to 1 and the others to 0. Moreover, θ represents the set of parameters and λ is a regularization parameter.

IV. Experiments for Sentence Classification

We mainly list and discuss detailed experiment settings and results in this section. The hyperparameters are displayed in Table I. Variants of our model are further compared and analyzed to demonstrate the robustness of the system at the end of this section.

A. Dataset and Data Augmentation

1) ***Dataset****:* The relevant models are tested on five datasets (SST-2 derives from SST-1).

- **SST-1** Stanford Sentiment Treebank [36]: It includes fine grained sentiment labels (very positive, positive, neutral, negative and very negative) for 215,154 phrases in the parse trees of 11,855 sentences extracted from movie reviews. In our experiments, we do not involve the sentiment information for phrases and only use the 11,855 sentence-level labels since our goal is sentence classification. We use the same way to split the dataset as in [36] into train (8544), dev (1101), and test splits (2210) and we also use the classification accuracy to measure the performances.
- **SST-2** With neutral reviews removed from SST-1, SST-2 only aims to classify a review as binary labels (negative, positive). The dataset is split into train (6920) , dev (872), test (1821) sets accordingly.
- **TREC** TREC question dataset (it is composed of train set (5452) and test set (500).)—the task involves classifying a question into six types (abbreviation, description, entity, human, location, numeric value) [37].
- **Subj** The aim is to classify sentences as either subjective or objective. This dataset is composed of 5000 instances of each [38]. We report the result of 10-fold cross validation as baseline systems did.
- **Yelp** The dataset is a 5-classes review dataset from Yelp [39], which contains millions of reviews. For this dataset, we follow all the settings in [43]. We parsed short reviews (less than 60 words) from the 200 most frequently reviewed restaurants. Also, we undersampled positive and very positive reviews as the reviews are skewed toward the positive end. We evaluated our model using 10-fold cross validation.

2) ***Data Augmentation****:* As is widely acknowledged, appropriate data augmentation mechanisms are expected to decrease generalization error in the field of deep learning. Before expand the dataset, we are obliged to figure out invariance properties of the specific task. To deal with texts, we have no similar way as processing images like rotating the images or adjusting the resolution. The exact order of words must be preserved for vital syntactic and semantic meaning. It seems that using human paraphrasing would have constituted an ideal approach of doing data augmentation, but such heavy labor is unaffordable and unefficient.

As a result, we acquired the method of replacing words or phrases with synonyms [40]. An English thesaurus, obtained from `mytheas` component in LibreOffice[1] was used in experimenting data augmentation. Ignoring polysemous phenomenon in English, every synonym to a word or phrase was selected according to the most frequent meaning, which simplified the task. Note that the process of replacement was determined by a geometric distribution: $P[t] = p^t$, where t was a parameter to represent replacement times.

To be specific, we display a small part of results of synonym replacement in Table II. Those samples are from Stanford Sentiment Treebank dataset, while the case is similar for TREC and subj datasets. The first and second samples show the one-word-level replacement. The third sample explains that replacement can be performed on multiple words (*affair vs. matter and life vs. being*). Further more, the fourth sample belongs to the case of phrase-level replacement. Last but not least, the operation does not promise a logical outcome, as we can see in the last sample (*place vs. topographic*). Above all, another point to underline is that the times of augmentation should be carefully decided, since the performance of

[1] `http://www.libreoffice.org/`

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE II
THE EXAMPLES OF SYNONYM-REPLACEMENT BASED ON STANFORD SENTIMENT TREEBANK DATASET.

Samples from Stanford Sentiment Treebank	*Replaced with*
The Rock is destined to be the 21st Century 's new " Conan " and that he 's going to make a splash even greater than Arnold Schwarzenegger, Jean-Claud Van Damme or Steven Segal.	*larger*
More timely than its director could ever have dreamed, this quietly lyrical tale probes the ambiguous welcome extended by Iran to the Afghani refugees who streamed across its borders, desperate for work and food.	*drawn*
... the tale of her passionate, tumultuous affair with Musset unfolds as Sand's masculine persona, with its love of life and beauty, takes form	*matter & being*
Enjoyably dumb, sweet, and intermittently hilarious – if you 've a taste for the quirky, steal a glimpse.	*pleasantly dull*
If you sometimes like to go to the movies to have fun, Wasabi is a good place to start.	*topographic*

system would be influenced by the operation largely. Over-augmentation might cause over-fitting of the system and the subsequent accuracy deceasing, while insufficient augmentation could not be powerful enough to make a difference.

B. Implementation Details

1) ***Pre-trainied Word Vectors****:* It is commonplace to initialize word vectors with those collected from an unsupervised neural language model. We used `word2vec` [34] vectors trained on 100 billion words from Google News and now publicly available. Continuous Bag-of-Words (cBoW) was used to train the words, yielding out vectors with dimensionality of 300. Difference of words' meaning would be reflected by euclidean distance in vector space. Note that words not contained in this pre-trained set were initialized randomly.

2) ***End2end Structure****:* We used `word2vec` to initialize the word vectors, but we also regarded word vectors as a part of parameters. So they would be updated simultaneously during the training to reduce the cost. Important as this action was, prominent improvement would be observed compared with non-end2end structure.

3) ***Parameter Initialization****:* Here we mainly refer to the initialization of convolutional kernels. Based on the method that sentences are split into overlapped phrases, we initialized the 3D kernels with uniform distribution only on two dimensions except the dimension of sequential length, which might be understood as 3D kernels were treated as several 2D kernels concatenated in third dimension. The special step was vital to training or the classification accuracy might be stuck from the beginning. Since we only utilized uniform distribution in this task, other common distributions like Gaussian distribution might lead to better result, which is worth further exploration.

4) ***Activation Function****:* Instead of using traditional hyperbolic tangent function ($tanh$), We selected rectified linear units (ReLu) as our activation function. It exhibited prominent ability of inhibiting varnishing gradient, making the training of deep network less difficult. Furthermore, it introduced sparsity to the network, which weakened interdependence of parameters and avoided over-fitting effectively. Our experiment process proved that ReLu made the accuracy converge faster and better. Besides, ReLu, the rectified linear units, would produce activation value of zero if input is zero, which is helpful to eliminate boundary effects caused by the great variability of sentence length [46].

5) ***Training Method****:* We conduct our experiments based on a open-source Python library Theano [47]. We observe that employing stochastic gradient descent (SGD) frequently leads to local minima on the error. Therefore we use RMSProp instead [48], which combines the idea of only using the sign of the gradient with the idea of adapting the step size separately for each weight. This significantly accelerates the training processing.

C. Experiment Results

In this section, we display the accuracies of our architecture, together with several state-of-the-art competitor networks on SST-1, SST-2, TREC and subj datasets. Detailed analyses are given for better understanding. An interesting fact is that we haven't observed significant increases on accuracies after data augmentation (less than 0.2% on the whole), thus we merely have performed augmentation to SST.

1) **Competitor Models:** As shown in Table III, we compare our results with following models. Note that all the methods used the same way to split the corresponding dataset into train/dev/test sets.

- **CNN-MC:** Convolutional neural network proposed by [45] with little hyperparameter tuning and static vectors.
- **RNTN:** Recursive Neural Tensor Network produces the hidden vector of the parent node by combining two descendant nodes, with the same, tensor-based composition function for all nodes [36].
- **MGNC-CNN:** Multi-Group Norm Constrained CNN is a scalable CNN which extracts features from three sets of word embeddings independently, and then joins these at the penultimate layer to form a final representation for classification [35].
- **MVCNN:** MVCNN is a CNN with multichannel variable-size convolution [49]. It combines various versions of pre-trained word embeddings, as well as extracting features of multi-granular phrases with variable-size convolution filters.
- **BLSTM-2DP:** Bidirectional LSTM with two-dimensional max-pooling [8] applies two-dimensional (2D) pooling operation over the two dimensions, which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE III
ACCURACIES (MEAN (MIN, MAX)) OF DIFFERENT MODELS ON THE TEST SET OF SST, TREC AND SUBJ AT SENTENCE LEVEL.

Algorithm	SST-1	SST-2	TREC	subj	Yelp
CNN-MC	47.4%	88.1%	92.2%	93.2%	–
RNTN	45.7%	85.4%	–	–	–
MGNC-CNN	48.65%	88.35%	95.52%	94.11%	55.8%
MVCNN	49.6%	89.4%	–	93.9%	–
BLSTM-2DP	50.5%	88.3%	94.8%	94.0%	–
S-LSTM	48.9%	81.9%	–	–	–
GRA	**51%**	87.9%	–	–	58.1%
3D-SPP CNN	50.8% (50.4%, 51.2%)	**89.5%** (89.0%, 90.3%)	**95.8%** (94.8%, 96.2%)	**94.5%**	**61.7%**

may sample more meaningful features for sequence modeling task

- **GRA:** a 'Gated Representation Alignment' (GRA) model [43] that blends a phrase focused Convolutional Neural Network (CNN) approach with sequence-oriented Recurrent Neural Network (RNN).
- **S-LSTM:** A model [7], in which a memory cell can reflect the history memories of multiple child cells or multiple descendant cells in a recursive process, has potentials of avoiding gradient varnishing, and hence may model long-distance interactions over trees.

2) **Results Analysis:** We show the accuracy performance of all methods in Table III. For SST-1, SST-2 and TEC datasets, to ensure that the propose methods performance improvement is not due to chance, we repeat the experiments for 10 times and show the mean (minimal, maximum) results among the 10 experiments. For subj and Yelp datasets, considering we have conducted the 10-fold cross validation strategy, we directly show the mean accuracy of 10 fold testing. Observe that the mean accuracy of our 3D-SPP-CNN ($k = 3$) outperforms other state-of-the-art baselines on 4 of 5 datasets. In particular, although our architecture is slightly inferior to GRA on SST-1 dataset, it achieves higher accuracy over other models in SST-2, TREC, subj, Yelp datasets by up to 7.6%, 3.6%, 1.3%, and 5.9% respectively. We can see our method can gain the respectful improvements in almost all the datasets, and the maximum results are even higher than mean results about 1% in all the datasets. S-LSTM [7] extends conventional chain-structured long short-term memory to tree structures, which are suitable for modeling long distance interactions. GRA [43], as a hybrid model, actually employs CNN to identify influential n-grams of different semantic aspects at first stage, which are then selected by RNN according to relevance. However, the most vital part of this model should be the soft-alignment layer between CNN and RNN, also viewed as an attention framework [50] to give weights for phrase vectors. Considering the the difficulty of data augmentation, we add limited data augmentation examples in all the experiments. The mean accuracy of the 3D-SPP CNN trained without data augmentation in SST-1, SST-2, TREC, subj, and Yelp are 50.75%, 89.4%, 95.72%, 94.26%, and 60.5% respectively. The effect of data augmentation is quite limited unless a more powerful data augmentation technology for natural language processing is designed.

TABLE IV
CONTRAST OF DIFFERENT CHOICES OF k. NOTE THAT THE CASE $k = 3$ ACHIEVES THE BEST PERFORMANCE.

Algorithm	SST	TREC	subj
3D-SPP CNN($k = 2$)	48.5%	94.2%	92.8%
3D-SPP CNN($k = 3$)	**50.8%**	**95.8%**	**94.5%**
3D-SPP CNN($k = 4$)	49.1%	94.5%	93.0%
3D-SPP CNN($k = 5$)	47.8%	92.1%	92.2%

TABLE V
CONTRAST BETWEEN THE INTEGRAL MODEL AND THE DEGENERATED MODEL WITHOUT N-GRAMS AS INPUT

Algorithm	SST	TREC	subj
3D-SPP CNN($k = 3$)	**50.8%**	**95.8%**	**94.5%**
SPP CNN	48.6%	94.0%	92.6%

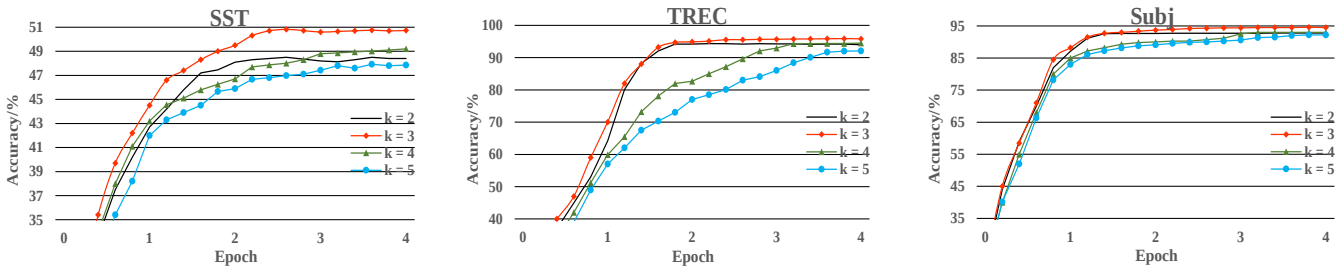
On the contrary, Our 3D-SPP CNN is design for exploiting phrase-level information as well as mitigating adverse impacts accompanied by length variability. As the interdependence of words generally weakens with distance increasing, the form of phrase reflects this kind of relation most straight and intensely. The superior performance of our architecture indicates that it can makes the best use of vital local dependences and achieves eminent classification accuracies without employing hybrid structure and attention mechanism.

D. Impacts of Model Architectures

We explore the effects of some hyper-parameter settings in this section. Distinct differences are observed on challenging SST-1 dataset. As the last section, to ensure that the propose methods performance improvement is not due to chance, we repeat the experiments for 10 times and show the mean results among the 10 experiments.

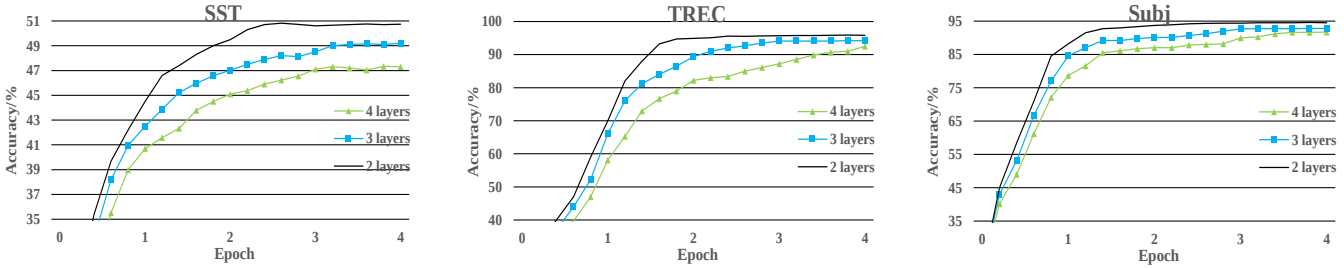
1) ***N-grams and Convolutional Layer****:* We may make some comparison based on experimental results to illustrate. The 3-layer CNN-MC with filter windows of 3, 4, 5 [45], achieves the accuracy of 47.4%, while our 2-layer tri-gram model without SPP layer reaches 48.5%. Besides, the contribution of tri-grams to the score in the relation classification task has also been displayed [4]. Reasonably, we expect the utilization of n-gram and according 3D convolution to enhance the efficiency of capturing correlations within phrases.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing



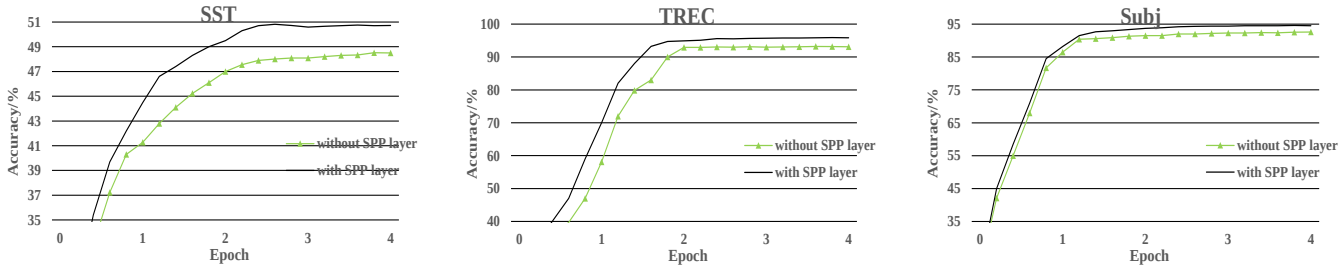
(a) (b) (c)

Fig. 7. The convergence performances of models included in Table IV. Experiments are conducted on SST, TREC and subj respectively, with phrase length k chosen from $[2, 3, 4, 5]$. Although distinctive difference can only be observed on Fig.7(a), it's still solid to conclude that $k = 3$ curve achieves the best accuracy on these datasets.



(a) (b) (c)

Fig. 8. The comparison of performances of our proposed system with various depth. Convolutional layers are set as $[2, 3, 4]$ in order to explore the best network depth for our task. As mentioned in Table VI, 2-layer system is substantiated to be the best setting in terms of the overall training situation on SST, TREC and even subj.



(a) (b) (c)

Fig. 9. The study on the effect of Spatial Pyramid Pooling layer. SPP layer has been proven to produce more discriminative representation to enhance classification accuracy on above datasets.

The detailed results are included in Table V to study the effects of using n-gram as input. SPP CNN is the degenerated version of our 3D architecture, which takes in four chunks of individual words, while the rest setting remains the same.

*2) **Choose Different** k:* To investigate the setting of k, we compare different models where k is set as $2, 3, 4, 5$ respectively, and show these results in Table IV. As observed from this table, $k = 3$ achieves the best accuracy, outperforming the closest result by more than 1%, which can be explained from two aspects.

Firstly, given that we already fix the size of convolutional filters and the output length as 4, the case of $k = 3$ simply suits the above setting best. Specifically, inputs become $[18 - k, 25 - k, 45 - k, 54 - k]$ after choosing a number for k. Then we require to adjust the combination of pooling rates in two pooling layers, making the network match the different length. Ranging from 2 to 5, case of $k = 3$ makes the model perform the best.

Here, we also offer a general rule based on experimental experiences that the pooling rates in the first pooling layer should better be small, i.e. no more than 3. The ratio in the second pooling layer should be configured to a larger value, but this should not exceed 5. The purpose of the selection rule to maximize the extraction important features while minimizing information loss.

In addition, we observe that $k = 3$ constitutes the best choice for length of phrase. This suggests that 3-word-phrase contains vital interdependency for our sentiment classification. Further, it is safe to speculate that the interdependence within 2-word-phrase is less important (from trivial experiences, 2-word-phrase usually fails to carry complete local information). Similarly, cases of $k = 4$ and $k = 5$ might provide redundant interdependence, causing the ratio of vital features decreasing to some extent.

*3) **Converging Time**:* The convergence behaviors during training are shown in Figure 7. There are four curves in total corresponding to $k \in [2, 3, 4, 5]$ and all the models use the same training configuration (including learning rate, batch-size

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE VI
COMPARISON BETWEEN 3D-SPP CNN($k = 3$) WITH DIFFERENT NUMBER OF CONVOLUTIONAL LAYERS. THE CASE OF TWO CONVOLUTIONAL LAYERS ACHIEVES THE BEST PERFORMANCE.

Algorithm	SST	TREC	subj
3D-SPP CNN(2 layers)	**50.8%**	**95.8%**	**94.5%**
3D-SPP CNN(3 layers)	48.9%	94.1%	92.8%
3D-SPP CNN(4 layers)	47%	92.5%	91.7%

TABLE VII
STUDY ON THE EFFECT OF SPATIAL PYRAMID POOLING LAYER. * DENOTES THAT THE SPP LAYER IS REMOVED FROM ORIGINAL NETWORK, WHILE THE OTHER SETTINGS REMAIN THE SAME.

Algorithm	SST	TREC	subj
3D-SPP CNN	**50.8%**	**95.8%**	**94.5%**
3D-SPP CNN*	48.5%	93.1%	92.6%

and so on). We observe that the convergence time increases with the increment of k. In particular, the $k = 2$ case converges in the second training epoch while $k = 3$ case starts to converge after about an epoch later. This is indeed reasonable, as the increment of k would enlarge the size of input, requiring more computation within a epoch of training. Admittedly, our models generally take longer to converge than traditional 2D CNNs, due to the impact of Our 3D architecture.

*4) **Various Depth of Networks**:* Table VI shows the various settings of numbers of convolutional layers and their accuracy. Observe prediction accuracies drops with depths of the architectures. It is acknowledged that increasing the depth of neural network does not necessarily guarantee better accuracy. For example, the performance of VGG network will go down when the number of layers is over 20 [51]. Our intuition is that deeper architectures complicate the optimized function, making it prompt to be stuck in local minima (Fig. 8). This phenomenon may also be resulted by overfitting. Since 3D architecture has more complex operations over 2D convolutional layers, employing this complicated model increases the risk of over-fitting.

*5) **Spatial Pyramid Pooling Layer**:* We conclude our experiments by examining the impact of SPP layer. SPP layer is designed for exploiting spatial information in CV tasks [6]. It is different to apply it into sentence classification task compared to the original SPP in vision tasks. We modify SPP layer to 3D phrase pooling layers to produce the most active features, and then concatenate them with original vector to obtain a more discriminate sentence-level representation. In Table VII, 3D-SPP network is with 2 convolutional layers and phrase-length of "k = 3".

The existence of 3D phrase pooling layers layer seems to have improved classification accuracy significantly on SST, TREC and Subj datasets according to Table VII, with an average improvement of more than 2% (Fig. 9). We also perceive that the phrase-level pooling method proposed can efficiently avoid redundancy as well as choose vital features.

V. EXPERIMENTS FOR RELATION CLASSIFICATION

We also conduct auxiliary experiments for the task of relation classification to demonstrate our model's capability as an efficient feature extractor, which aims to identifying the semantic relation holding between two tagged entities in a sentence. For instance, in the sentence "The size of a tree crown is strongly ... ", the goal would be to automatically categorize the relationship of the word pair as a Component-Whole type.

Precise relation classification boosts accurate sentence interpretations, discourse processing, and higher-level NLP tasks [52]. However, we may depict the same kind of relation with various ways, and this complex variability can be lexical, syntactic, or even pragmatic in nature. Therefore, traditionally algorithms mainly focused on feature- or kernel- based approaches, many of which relied on extensive manual feature engineering or use of external resources [53], [54], [55]. Recent years have witnessed a move towards deep architectures, enabling the model to learn relevant representations without rich prior knowledge. A number of convolutional neural network (CNN), recurrent neural network (RNN), and other neural architectures have been proposed for relation classification [56], [57]. Nevertheless, our model is capable of identify critical cues within n-grams, while does not require an external dependency parser as many neural systems do.

To predict semantic relation using the proposed architecture. The way of forming input representation is slightly different from that in sentence classification. We firstly incorporate word position embedding (WPE) to o reflect the relative distances between the i-th word to the two marked entity mentions [56], [57]. Every relative distance is mapped a randomly initialized position vector in R^p, where p is a hyper-parameter. So the final word embedding for the i-th word becomes $z_i = [(x_i), (x_{i,1}), (x_{i,2})]$, where $x_i \in R^d$ is the word vector, $x_{i,1}$ and $x_{i,2} \in R^p$ are the first and second relative position vectors respectively. The rest model configuration can be referred by Section V.

A. Dataset and Metric

We conduct experiments on commonly used SemEval-2010 Task 8 dataset [52], employing the macro-averaged F1-score to evaluate different methods. This dataset consists of 8,000 sentences for training and 2,717 sentences for testing. For both relation directions, the dataset consists of nine relation types. And there exists a Other class. Hence, the dataset has 19 classes in total for semantic relations. Also, we have calculate the sentence length for this dataset, and there are only 19 sentences in training set and 6 sentences in testing set which have more than 53 words. Therefore, we divide sentences of those dataset into four sections, and the lengths of each are $[17, 24, 44, 53]$. We follow all the setting in Table I as last experiment part.

B. Evaluation

We compare our model with previous state-of-the-art methods, including those relying heavily on prior knowledge.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2852502, IEEE/ACM Transactions on Audio, Speech, and Language Processing

- SVM: Support Vector Machine with [58] a number of features that capture the context, semantic role affiliation, and possible pre-existing relations of the nominals.
- RNN: A recursive neural network (RNN) model that learns compositional vector representations for phrases and sentences of arbitrary syntactic type and length [59].
- MVRNN: A combination of matrix-vector representations with a recursive neural network, so that it can learn both the meaning vectors of a word and how that word modies its neighbors(via its matrix) [59].
- FCM: A compositional model for deriving sentence-level and substructure embeddings from word embeddings [60].
- DRNNs: The deep recurrent neural network can explore the representation space in different levels of abstraction and granularity [61].
- CNN+softmax: A convolutional deep neural network, where position features (PF) are successfully proposed to specify the pairs of nominals to which we expect to assign relation labels [56].
- CR-CNN: A convolutional neural network that performs classification by ranking [57].
- Ours*: The two-layer and tri-gram version of the proposed system but without SPP mechanism.
- Ours: The two-layer and tri-gram version of our proposed system.

According to Table VIII, our model outperforms other state-of-art competing methods, without relying on any manually designed features. Specifically, the proposed 3D-SPP network achieves the macro-averaged F1-score of 87.2%, thus exceeding not only the original winner of the SemEval task, a SVM-based approach (82.2%) depending on rich feature engineering, by 5.0%, but also the well-known CR-CNN (84.1%) with a margin of 3.1%, and recent DRNNs (85.8%) by 1.4%.

To demonstrate our system as an efficient feature extractor, firstly, tri-grams can provide rich contextual information for discerning patterns encoded in complex-structured sentences, which is supported by results in our experiment on UCF101 ($k = 3$ has been proved to be the best setting for capturing the relation of words in sentences). Secondly, SPP improves the capability of capturing internal dependencies significantly, based on the fact that the loss of 1.8% in accuracy if SPP mechanism is removed from our original network.

TABLE VIII
EXPERIMENTAL RESULTS FOR RELATION CLASSIFICATION, WHERE * REFERS TO THE REMOVAL OF SPP MECHANISM FROM OURS.

Model	F1
Manually Engineering methods	
SVM	82.2
Dependency Models	
RNN	77.6
MVRNN	82.4
FCM	83.0
DRNNs	85.8
End-To-End Methods	
CNN+softmax	82.7
CR-CNN	84.1
Ours*	85.4
Ours	**87.2**

VI. CONCLUSION

The sentence classification problem is becoming increasingly essential in the NLP research area. In this paper, we propose a novel deep learning architecture which combines 3D convolutional layer and Spatial Pyramid Pooling layer tailored to this problem. In particular, we restructure input sentences to a concatenation of phrases rather than naive words sequences. We extend traditional CNNs to 3D phrase-level CNNs to enable model to capture phrase-level features more effectively. This also avoids side effects of sentence length variability. To eliminate the boundary effects induced by great length variety, we upgrade Spatial Pyramid Pooling (SPP) to a 3D phrase pooling layer, which allow models to accept flexible length of input and reduce noise introduced by zero padding. Evaluation on 4 benchmark sentence classification datasets (i.e. Stanford Sentiment Treebank [36], TREC [37] and Subj [38]) demonstrate that our proposal system achieves superior accuracies on 3 dataset, outperforming state-of-the-art deep learning models. Auxiliary experiments on SemEval-2010 Task 8 dataset [52] can consolidate the model's effectiveness as a universal feature extractor.

We believe that the potential of employing CNNs to address NLP problems remains unexplored. CNNs should be agnostic to word-level or character-level inputs. The difference between two representation types could be studied on a series of large-scale datasets. Moreover we intend to adapt our model to other language processing tasks including morphologically rich languages (e.g., Russian) or languages without alphabetic writing systems (e.g., Chinese).

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation of China with Grants 61401169.

REFERENCES

[1] N. Kalchbrenner, E. Grefenstette and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. Association for Comput. Linguist.*, pp. 352-357, 2014.
[2] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using LSTM for region embeddings," in *Proc. Int. Conf. Mach. Learn.*, pp. 526-534, 2016.
[3] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Int. ACM Sigir Conf. on Research and Development in Inf. Retrieval.*, pp. 83-92, 2014.
[4] L. Wang, C. Zhu, M. G. De Melo and Z. Liu, "Relation classification via multi-level attention CNNs," in *Proc. of the 54th Annual Meeting of the Association for Comput. Linguist.*, PP. 1298-1304, 2016.
[5] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 221-231, 2013.
[6] K. He, X. Zhang, S. Ren and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. IEEE Eur. Conf. Comput. Vis.*, pp. 346-361, 2014.

[7] X. Zhu, P. Sobhani and H. Guo, "Long short-term memory over recursive structures," in *Proc. Int. Conf. Mach. Learn.*, pp. 1604-1612, 2015.

[8] 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," *CoRR*, [Online]. Available: https://arxiv.org/abs/1611.06639, 2016.

[9] E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," *IEEE Comput. Int. Mag.*, vol. 9, no.

[10] I. Sutskever, O. Vinyals and Q. Le, "Sequence to sequence learning with neural networks," *CoRR*, [Online]. Available: https://arxiv.org/abs/1409.3215, 2014.

[11] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. Association for Comput. Linguist.*, pp. 295-302, 2002.

[12] J. Zhang, S. Liu, M. Li, M. Zhou and C. Zong, "Mind the gap: machine translation by minimizing the semantic gap in embedding space," in *Proc. Association for Adv. Artif. Intell.,* pp. 1657-1664, 2014.

[13] H. AlMubaid, "A learning-classication based approach for word prediction," *Int. Arab J. Inf. Technol.,* vol. 4, no. 3, pp. 264-271, 2007.

[14] N. Garay-Vitoria and J. Abascal, "Modelling text prediction systems in low- and high-inected languages," *Comput. Speech Lang.,* vol. 24, no. 2, pp. 117-135, 2010.

[15] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp and G. Weikum ,"Natural language questions for the web of data," in *Proc. 2012 Joint Conf. Empir. Meth. Nat. Lang. Process. and Comput. Nat. Lang. Learn.,* pp. 379-390, 2012.

[16] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classication of reviews," in *Proc. Association for Comput. Linguist.,* pp. 417-424, 2002.

[17] M. Taboada, J. Brooke , M. Toloski, K. Voll and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. linguist.,* vol. 37, no. 2, pp. 267-307, 2011.

[18] M. Thelwall, K. Buckley and G. Paltoglou, "Sentiment strength detection for the social web," *J. Amer. Soc. Inf. Sci. Technol.,* vol. 63, no. 1, pp. 163-173, 2012.

[19] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up?: Sentiment classication using machine learning techniques," in *Proc. Empir. Meth. Nat. Lang. Process.,* pp. 79-86, 2002

[20] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classication," in *Proc. Association for Comput. Linguist.,* pp. 90-94, 2012.

[21] R. Socher, C. D. Manning and A.-Y. Ng, "Learning continuous phrase representations and syntactic parsing with recursive neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1-9, 2010.

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

[23] L. Dong, F. Wei, K. Xu, S. Liu and M. Zhou, "Adaptive multi-compositionality for recursive neural network models," *IEEE Trans. Audio, Speech, Lang. Process.,* vol. 24, no. 3, pp. 422-431, 2016.

[24] K. Tai, R. Socher and D. C. Manning, "Improved semantic representations from tree-structured long short-term memory Networks," In *Proc. Association for Comput. Linguist.*, pp. 1556-1566, 2015.

[25] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola and E. Hovy, "Hierarchical attention networks for document classification," in *North Amer. Chapter of the Association for Comput. Linguist.*, pp. 1480-1489, 2016.

[26] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. Association for Comput. Linguist.*, pp. 207-212, 2016.

[27] Y. Chen, B. Perozzi, R. Al-Rfou, and S. Skiena, "The expressive power of word embeddings," *CoRR*, [Online]. Available: arXiv:1301.3226, 2013.

[28] A. Mnih and G. Hinton, "A scalable hierarchical distributed language model, in *Proc. Adv. Neural Inf. Process. Syst.* pp. 1081-1088, 2009.

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

[30] F. Hill, K. Cho, S. Jean, C. Devin and Y. Bengio, "Not all neural embeddings are born equal," *CoRR*, [Online]. Available: arXiv:1410.0718, 2014.

[31] C. Santos, M. Tan, B. Xiang and B. Zhou, "Attentive pooling networks," *CoRR*, [Online]. Available: https://arxiv.org/abs/1602.03609v1, 2016.

[32] M. Tan, C. Santos, B. Xiang and B. Zhou, "Lstm-based deep learning models for nonfactoid answer selection," *CoRR*, [Online]. Available: https://arxiv.org/abs/1511.04108, 2015.

[33] T. Rocktaschel, E. Grefenstette, K. Hermann, T. Kocisky and P. Blunsom, "Reasoning about entailment with neural attention," *CoRR*, [Online]. Available: https://arxiv.org/abs/1509.06664, 2015.

[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, [Online]. Available: https://arxiv.org/abs/1504.05070, 2015.

[35] Y. Zhang, S. Roller and B. Wallace, "A simple approach to exploiting multiple word embeddings for sentence classification," *CoRR*, [Online]. Available: https://arxiv.org/abs/1603.00968, 2016.

[36] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng and C. Potts, "Recursive deep models for semantic compositionality Over a sentiment treebank," in *Proc. Empir. Meth. Nat. Lang. Process.*, pp. 1631-1642, 2013.

[37] X. Li and D. Roth, "Learning question classiers," in *Proc. Association for Comput. Linguist.*, pp. 1-7, 2002.

[38] B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. Association for Comput. Linguist.*, pp. 271-278, 2004.

[39] T. Duyu, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. on Empirical Methods in Natural Language Processing.*, pp. 1422-1432, 2015.

[40] X. Zhang, J. Zhao and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 649-656, 2015.

[41] J.Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," in *North Amer. Chapter of the Association for Comput. linguist.*, pp. 515-520, 2016.

[42] S. Lai, L. Xu, K. Liu and J. Zhao, "Deep recursive neural networks for compositionality in language," in *Proc. Association for Adv. Artif. Intell.*, pp. 2267-2273, 2015.

[43] S.T. Hsu, C. Moon, P. Jones and N.F. Samatova, "A hybrid CNN-RNN alignment model for phrase-aware sentence classification," in *Proc. of the 15th Conf. of the Eur. Chapter of the Association for Comput. linguist.*, PP. 443-449, 2017.

[44] X. Ouyang, S. Kawaai, E. G. H. Goh, S. Shen, W. Ding, H. Ming, and D. Y. Huang, "Audio-visual emotion recognition using deep transfer learning and multiple temporal models," in *Proc. of the 19th ACM International Conference on Multimodal Interaction*, pp. 577-582, 2017.

[45] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Empir. Meth. Nat. Lang. Process.*, pp. 1746-1751, 2014.

[46] B. Hu, Z. Lu, H. Li and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 2042-2050, 2014.

[47] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin , R. Pascanu , G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. of the Python for Scientific Comput. Conf.*, 2010

[48] T. Tijmen and G. Hinton, "A logic of implicit and explicit belief," *the Association for the Advance of Artificial Intelligence*, pp. 198-202, 2012.

[49] W. Yin and H. Schütze, "A simple approach to exploiting multiple word embeddings for sentence classification," *CoRR*, [Online]. Available: https://arxiv.org/abs/1603.00968, 2016.

[50] D. Bahdanau, K. cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, [Online]. Available: arXiv:1409.0473, 2014.

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

[52] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha, S. Padó, M. Pennacchiotti,L. Romano, and S. Szpakowicz, "SemEval-2010 task 8: multi-way classification of semantic relations between pairs of nominals," in *Proc. of the 5th Inter. Workshop on Semantic Evaluation, ACL*, pp. 33-38, 2010.

[53] R. J. Mooney and R. C. Bunescu, "Subsequence kernels for relation extraction," in *Adv. in Neural Inf. Process. Syst.,* pp. 171-178, 2005.

[54] N. Kambhatla, "Combining lexical,syntactic, and semantic features with maximum entropy models for extracting relations," in *Proc. of the 42nd Annual Meeting of the Association for Comput. Linguist.*, pp. 22, 2004.

[55] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," in *Proc. of the Conf. on Human Lang. Tech. and Empirical Methods in Natural Lang. Process. , ACL*, pp. 724-731, 2005.

[56] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, "Relation classication via convolutional deep neural network," in *COLING*, pp. 2335-2344, 2014.

[57] C. N. dos Santos, B. Xiang and B. Zhou. "Classifying relations by ranking with convolutional neural networks," In *Proc. of the 53rd Annual*

Meeting of the Association for Comput. Linguist. and the 7th Inter. Joint Confer. on Natural Lang. Process., volume 1, pp. 626-634, 2015.

[58] B. Rink and S. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," in *Proc. of the 5th Inter. Workshop on Semantic Evaluation, ACL*, pp. 256-259, 2010.

[59] R . Socher, B. Huval, C. D. Manning and A. Y. Ng. "Semantic compositionality through recursive matrix-vector spaces," in *Proc. of the 2012 Joint Conf. on Empirical Methods in Natural Lang. Process. and Comput. Natural Lang. Learning*, pp. 1201-1211, 2012

[60] M. Yu, M. Gormley and M. Dredze. "Factor-based compositional embedding models," in *NIPS Workshop on Learning Semantics*, pp. 95-101, 2014.

[61] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Ya. Lu and Z. Jin. "Improved relation classification by deep recurrent neural networks with data augmentation," *CoRR*, [Online]. Available: https://arxiv.org/abs/1601.03651, 2016.

Xi Ouyang received his B.S degree at the School of Electronic Information and Communications of Huazhong University of Science and Technology, and a M.S. degree in the Department of Electronics and Information Engineering from HUST, Wuhan, China, in 2015 and 2018, respectively. He was a research intern in Panasonic R&D Center Singapore in 2016-2017, and worked on deep learning research. Currently he is purchasing the PhD at Med-X Research Institute, School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include: deep learning, medical image analysis, and natural language processing.

Kang Gu received his B.S degree at the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, P.R. China in 2018. Currently he is purchasing the PhD at Department of Computer Science, Dartmouth College, USA. His research interests are about natural language processing and the applications of deep learning.

Pan Zhou is currently an associate professor with School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, P.R. China. He received his Ph.D. in the School of Electrical and Computer Engineering at the Georgia Institute of Technology (Georgia Tech) in 2011, Atlanta, USA. He received his B.S. degree in the Advanced Class of HUST, and a M.S. degree in the Department of Electronics and Information Engineering from HUST, Wuhan, China, in 2006 and 2008, respectively. He held honorary degree in his bachelor and merit research award of HUST in his master study. He was a senior technical member at Oracle Inc, America during 2011 to 2013, Boston, MA, USA, and worked on hadoop and distributed storage system for big data analytics at Oralce cloud Platform. His current research interest includes: machine learning, big data analytics, privacy and security.