2018 International Conference on Artificial Intelligence and Big Data

Convolutional Neural Network with Pair-Wise Pure Dependence for Sentence Classification

Lingling Song, Yazhou Zhang, Yuexian Hou
School of Computer Science and Technology
Tianjin University
Tianjin, China
e-mail: songlingling_tju@163.com, yzhou_zhang @tju.edu.cn, yxhou@tju.edu.cn

Abstract—**Sentence classification has always been a crucial research topic in Natural Language Processing (NLP). Classical bag-of-words based models have a major limitation: the contextual information between words, which is the key to form meaningful semantic entities, is missing. Moreover, the semantic entities are not necessarily limited to syntactically valid phrases or named entities, but can be high-order association (also referred to as high-order *dependence*) patterns. To address this issue, in this paper, we propose PPD-CNN, a convolutional neural network (CNN) architecture with Pair-wise Pure Dependence (PPD) for sentence classification. Compared with the traditional CNN, our PPD-CNN (1) combines PPD pattern which is a couple of dependence words as strong un-separable high-level semantic entity and (2) extracts multi-granular semantic information, which treats PPD pattern as an input channel to capture the whole features and the original sentence as another input channel through variable-size convolution filters to catch all kinds of local features. With this design, our PPD-CNN can model the contextual information, which is important for grasping the word sense. The experimental results show that our approach significantly outperforms a wide range of baselines and state-of-the-art methods** *(Abstract)*

Keywords-CNN; pair-wise pure dependence; sentence classification

I. Introduction

Sentence classification is a fundamental task in natural language processing (NLP) and has attracted an increasing attention from both academia and industry. It aims at discovering diversified semantic information implied in the given natural language text. Therefore, a world of approaches mainly focus on extracting features for different text units and performing compositions over a variable-length sequences through machine learning or semantic rule based methods However, the choice of representations and features is a completely empirical process, driven by the intuition, experience and domain expertise [2]. With the development of deep learning, some researchers have designed effective neural networks to automatically generate useful low dimensional representations for capturing salient semantic and syntactic properties from the context and obtain a promising result on the sentence classification task [3].

Socher et al. proposed the Recursive Neural Network (Recursive NN) that has been proven to be efficient in terms of constructing sentence representations [4][5]. The Recursive NN captures the semantics of a sentence via a tree structure, which depends on well-performing parsers. Recurrent Neural Network (Recurrent NN) analyzes a sentence word by word, aiming at capturing the contextual information [6]. However, later words are more dominant than earlier words in RNN, even incorporating the gating mechanism and attention mechanism into the RNN variants (e.g. LSTM and GRU [29]), leading to a biased representation on sequential textual modeling. The traditional CNN systems usually implement a convolution layer with variable-size filters (i.e., feature detectors), in which the concrete filter size is a hyper parameter [26]. They essentially split a sentence into multiple sub-sentences by sliding windows, and then determine the sentence label by using the dominant label across all sub-sentences [7]. The underlying assumption is that the sub-sentence with that granularity is potentially good enough to represent the whole sentence. However, these sub-sentences just capture the local features, which is insufficient to model the whole features. Researchers have tried a lot of efforts to combine CNN with all kinds of RNN incorporating long distance information [8][9], but RNN's bias problem or the problem of relying on the parser tree still restrict them to achieve a better classification performance.

In [10], it has been proved that the words in the same association serve as context of each other and form a high-level semantic meaning whose distribution cannot be conditionally factorized, and the word association is defined as high-order pair-wise pure dependence (PPD). For example, a PPD pattern {*climate, conference, Copenhagen*} in a sentence, it implies an un-separable high-level semantic entity which cannot be fully explained as the random coincidence of, for example, the co-occurrence of "*Copenhagen*" and "*conference*" (which could be any other conferences in Copenhagen) and the occurrence of "*climate*".

Hence, combining CNN with PPD pattern seems a fascinating way for developing excellent sentence classification model, in which PPD can capture long distance context and extract integral additional un-separable high-level semantic effectively.

Contributions. (1) We propose PPD-CNN, a novel scalable CNN architecture that combines PPD pattern which is a couple of dependence words as strong un-separable high-

978-1-5386-6987-7/18/$31.00 ©2018 IEEE

level semantic entity. (2) Our model treats sentence and PPD pattern words as distinct input channels, thus generating corresponding feature vectors which are then concatenated at the classification layer. (3) Our model achieves state-of-art results on several datasets.

II. Related Work

Sentence classification has attracted an increasing attention in the last few decades [1]. Prior work has considered combining latent representations of words that capture syntactic and semantic properties for general NLP tasks [11]. As the classical sentence classification work using CNN model, Kim studied multichannel representation and variable-size filters [12]. Multichannel Variable-Size CNN (MVCNN) combined multiple word embeddings for sentence classification [13]. Le and Mikolov [14] initialized the representation of a sentence as a parameter vector, treating it as a global feature and combining this vector with the representations of context words to do word prediction.

In [10], the semantic entities are not necessarily limited to syntactically valid phrases or named entities. More generally, they can be high-order association (also referred to as high-order *dependence*) patterns, which are often beyond pairwise relations, for example, {“climate”, “conference”, “copenhagen”}. Such contextual high-order word association indicates that the words in the same association serve as context of each other and form a high-level semantic meaning.

For sentence modelling, a sufficient and unbroken meaning can not only supply the context, but also avoid the noise in integral context. Therefore, PPD is important to model integral context and contained the global long-term dependency. With the PPD, we can model the semantic information more explicitly.

It is well-known that the sentence is viewed as a mixture of words. In fact, a PPD pattern is also not limited to only one word. Different words will have different contributions to the final representation of both PPD patterns and sentences.

Hence, inspired by the studies of multi-channel CNN and the idea of word embeddings, and thus we consider combining PPD words as one channel of a multi-channel CNN.

Our model takes sentences and PPD pattern word embeddings as inputs. They are then treated as two separate ‘channels’, analogous to RGB channels in images. Note that this differs from ‘multi-channel’ models. We use different filters at the convolution layer on each word embedding matrix independently, whereas each filter would consider all channels simultaneously and generate a scalar from all channels at each local region in a traditional multi-channel approach.

The PPD-CNN model is composed of two parts which model the PPD pattern and sentence interactively. Taking word embeddings as input, we employ convolution layer to extract features for a PPD pattern and its corresponding sentence respectively. We use the max pooling to supervise the generation of the features to capture the most important information in the sentence and PPD pattern. With this design, the PPD pattern and sentence can influence the generation of their representations interactively. Finally, PPD pattern representation and sentence representation are concatenated as final representation which is fed to a softmax function for sentence classification.

III. Model Architecture

In this section, we first introduce the architecture of PPD-CNN model for sentence classification. Then, we describe the training details of PPD-CNN. The overall architecture of PPD-CNN model is shown in Figure 1.

A. Preliminaries

In this subsection, we describe the preliminaries of our model. First we initialize the input sentences and pre-trained word embeddings and thus we present how to mine the PPD pattern in the sentences. The process of mining PPD words is shown in Algorithm 1 for illustration.

Last, we generate the 3-order PPD pattern words of each sentence as another channels’ input based on the algorithm.

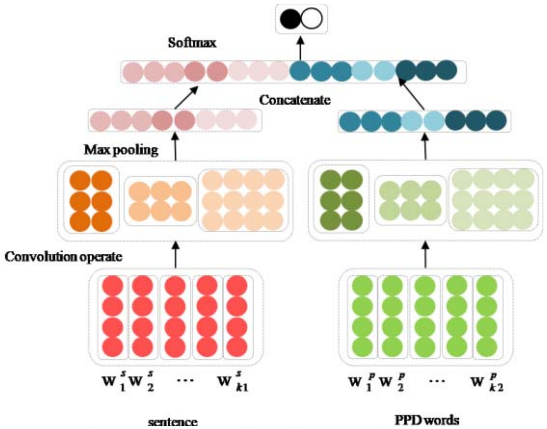

Figure 1. The overall architecture of PPD-CNN.

Algorithm 1 Mining PPD

```
$L_1 \leftarrow \emptyset$;
for every word $w$ where $df_w \geq min_f$ do
        $L_1 = L_1 \cup \{\{w\}\}$
while $L_{k-1}$ is not empty do
        $L_k \leftarrow \emptyset$
        for $a, b \in L_{k-1}$ do
            if $|a \backslash b| = 1$ AND $|b \backslash a| = 1$ then
                    if $a \cup b - a \cap b$ is $2-$order PPD then
                            $L_k \leftarrow L_k \cup \{a \cup b\}$
                            $k \leftarrow k + 1$;
                    $L \leftarrow L_2 \cup L_3 \cup \cdots \cup L_{k-1}$;
for $a, b \in L$ do
        if $a \subset b$ then
            $L \leftarrow L \backslash \{a\}$;
Return $L$;
```

Note that L_k ($k \in N$) is the set of all word patterns having k-order pure dependence. Also note that the test of two-order pure dependence could be an LLRT or Chi-squared test.

B. Model Description

Specifically, we first formalize the notation. We suppose that a sentence consists of n words $[w_s^1, w_s^2, \cdots, w_s^n]$ and a

PPD pattern has *m* words $[\mathrm{w}_p^1, \mathrm{w}_p^2, \cdots, \mathrm{w}_p^m]$. w denotes a specific word. To represent a word, we embed each word into a low dimensional real-value vector, called word embedding [15]. Then, we can get $\mathrm{w}^k \in R^d$, where k is the word index in the sentence or PPD, d means the dimensionality of word embeddings. Word embeddings can be regarded as parameters of neural networks or pre-trained from proper corpus via language model [16][17][18][19]. In our work, we choose the latter strategy.

Now, we use the multi-channels CNN networks to learn all kinds of features. We perform convolution operations which are different groups of filters $[\mathrm{w}_1^s, \mathrm{w}_2^s, \cdots, \mathrm{w}_{k1}^s]$, $[\mathrm{w}_1^p, \mathrm{w}_2^p, \cdots, \mathrm{w}_{k2}^p]$ to sentence and PPD respectively. For each $\mathrm{w}_i^s \in R^{h \times d}$ and $\mathrm{w}_i^p \in R^{h \times d}$, where h denotes 'height', we slide filter i across the sentence and PPD, considering 'local regions' of h adjacent rows at a time. At each local region, we perform element-wise multiplication and then take the element-wise sum between the filter and the (flattened) sub-matrix of sentence and PPD, producing a scalar. A filter slide in sentence resulting in a feature map vector $c_i \in R^{(s-h+1)\times 1}$.

$$c_i^s = f\left(w_i^s \otimes w_s + b^s\right), i = 1, 2, \cdots, k_1 \tag{1}$$

$$c_i^p = f\left(w_i^p \otimes w_p + b^p\right), i = 1, 2, \cdots, k_2 \tag{2}$$

where f is an activation function, which can be Tanh, Relu, Sigmoid and Iden etc. According to requirements, it may have multiple filter sizes, and multiple filters of each size may be introduced. These in turn generate corresponding feature maps $[\mathrm{c}_1^s, \mathrm{c}_2^s, \cdots, \mathrm{c}_{k1}^s]$ and $[\mathrm{c}_1^p, \mathrm{c}_2^p, \cdots, \mathrm{c}_{k2}^p]$ to sentence and PPD.

Then a max-pooling operation is applied to each feature map, extracting the most discriminating feature o_i from each feature map c_i .

$$o_i^s = \max\left(c_i^s\right), i = 1, 2, \cdots, k_1 \tag{3}$$

$$o_i^p = \max\left(c_i^p\right), i = 1, 2, \cdots, k_2 \tag{4}$$

Finally, we combine the sentence representation o^s and PPD representation o^p together to form a feature vector $o \in R^{\mathrm{k1+k2}}$, which is took as feature vectors through a softmax function for classification. We use a non-linear layer to map $o \in R^{\mathrm{k1+k2}}$ into the space of the targeted C classes. This process can be formulated as:

$$x = \tanh(\mathrm{w} \cdot o + b) \tag{5}$$

where *w* and *b* are the weight and bias respectively. The probability of labeling document with sentiment polarity $i(\mathrm{i} \in [1, \mathrm{c}])$ is computed by:

$$p_i = \frac{\exp(\mathrm{x}_i)}{\sum_{i=1}^{c} \exp(\mathrm{x}_i)} \tag{6}$$

The label with the highest probability is set as the final result.

C. Model Training

In PPD-CNN, we need to optimize all the parameters notated as Θ which are from the conv layer w_i^s , w_i^p , b^s , b^p , the softmax layer w , b and the word embeddings. The network is trained to minimize cross entropy with L_2 regularization of predicted and true distributions, which is defined as:

$$J = -\sum_{i=1}^{c} g_i \log(\mathrm{p}_i) + \lambda_r (\sum_{\theta \in \Theta} \theta^2) \tag{7}$$

where $g_i \in R^c$ denotes the ground truth, represented by onehot vector; $p_i \in R^c$ is the estimated probability for each class, λ_r is the coefficient for L_2 regularization.

A dropout operation is perform before the logistic regression layer [20]. The network is trained by back-propagation in mini-batches and the gradient-based optimization is performed using the AdaGrad update rule [21].

IV. EXPERIMENTS

A. Datasets

To demonstrate the effectiveness of the proposed method, we conduct the extensive experiments on the task of sentence classification using the following datasets.

- **MR**: It is a widely used dataset for sentiment analysis of movie reviews. The dataset consists of 5, 331 positive and 5, 331 negative reviews, each of which is a sentence. The task is to classify the reviews as positive or negative [22].
- **SUBJ**: Subjectivity is released by Pang and Lee [23]. The dataset contains 5, 000 subjective sentences and 5, 000 objective sentences. Classification involves detecting subjective or objective sentences.
- **CR:** Customer Review contains various digital products reviews from Amazon. Our task is to classify each of them as positive or negative [24].
- **SST-2**: Stanford Sentiment Treebank contains movie reviews but with train/dev/test splits provided and fine-gained binary labels [25].

MR, SUBJ, CR do not have fixed train and test splits, thus we run the experiments using 10-fold cross-validation for all comparative models.

B. Evaluation Metric

To evaluate the performance of sentence sentiment classification, we adopt the **Accuracy** metric, which is defined as:

$$Acc = \frac{T}{N} \tag{8}$$

where T is the number of correctly predicted samples, N is the total number of samples. Accuracy measures the percentage of correct predicted samples in all samples. Generally, a well performed system has a higher accuracy.

C. Hyperparameters Settings

In our experiments, we use the Glove tool to produce word embeddings. It is worth mentioning that we also train our embeddings using the Gensim API, and we find that it makes no difference from the Glove [19]. All out-of-vocabulary words and all weight matrices are initialized by sampling from the uniform distribution U(−0.0001, 0.0001), and all biases are set to zeros. The dimensionality of word embeddings is set to 300. **Tanh** is chosen as the nonlinear function in the convolution layer. The coefficient of L_2 normalization is set to 0.02, and the dropout rate is set to 0.5. The heights of filters are set to [3, 5] respectively.

D. Results and Discussion

In order to comprehensively evaluate the performance of CNN-PPD, we list some baseline approaches for comparison.

Random is a basic baseline method, which assigns random of sentiment polarity to each sample in the test set.

CNN is Convolutional Neural Network with multichannel proposed by Kim [12]. It is a model with two sets of word vectors. Each set of vectors is treated as a 'channel' and each filter is applied to both channels, but gradients are back propagated only through one of the channels. Hence the model is able to fine-tune one set of vectors while keeping the other static. Both channels are initialized with word2vec.

CNN variants are other methods rooting on the ground of CNN, such as dynamic convolutional neural network with k-max pooling (DCNN), dependency sensitive convolutional neural networks (DSCNN), multichannel variable-size convolutional neural networks (MVCNN), multi-group norm constraint CNN (MGNC-CNN), harnessing convolutional neural networks with logic rules (CNN-Rule), a CNN model which is initializing convolutional filters with semantic features (CNN+ n-gram) etc.

TABLE I. RESULTS COMPARISON WITH BASELINES (ACC)

Datasets/ Model	MR	SUBJ	CR	SST-2
Random	0.501	0.492	0.511	0.508
CNN [12]	0.815	0.934	0.843	0.872
DCNN [26]	-	-	-	0.868
DSCNN[9]	0.822	0.939	-	0.887
MVCNN [7]	-	0.939	-	**0.894**
MGNC-CNN [13]	-	0.941	-	0.883
CNN-Rule [27]	0.817	-	0.853	0.893
CNN+ n-gram [28]	0.821	0.937	**0.860**	0.890
CNN-PPD	**0.827**	**0.943**	0.858	**0.894**

Table 1 shows the performance comparison of CNN-PPD with other baselines. From Table 1, we can observe that, the Random method is the worst as we expected. All the other methods rooting on the ground of CNN get better classification results than the Random method, showing that CNN has powerful ability on automatically generating representations, thus can all bring performance improvement for sentiment classification.

Kim's CNN outperforms DCNN in terms of accuracy, since the former set various filter size, so it is necessary for our model to use many kind of filters.

Kim's CNN and DCNN methods get the worst performance of the rest neural network baseline methods, since it treats all of words in sentence equally. This also implies that the PPD work in [10] which points out PPD pattern can lead a better representation of sentence. It also illustrates that combining PPD words into CNN is efficient and is an important way for developing better sentence classification models.

MV-CNN and MGNC-CNN utilize multiple pre-trained embeddings as inputs. Both of them stably exceed Kim's CNN method. Because they not only use multichannel CNN, but also explore the combination of multiple public embedding versions to initialize words in sentences. This verifies that another channel whose input differing from the sentence channel can extract latent semantic information. Employing the multi-channel strategy seems a reasonable way to improve the performance. Hence, we use the PPD as another channel is effective.

Further, CNN-Rule integrates logic rules and CNN+n-gram encodes semantic features into the filters instead of initializing them randomly. It means that it is useful to make syntax and semantic rules in NLP. However, how to define excellent semantic rules is the key to extracting sentence features. It is natural for us to bring PPD features into CNN. Last, our model achieves the best accuracy results on most datasets. Compared with MVCNN and MGNC-CNN，our model uses different filters on each word embedding matrix independently at the convolution layer, where in a standard multi-channel approach each filter would consider all channels simultaneously. In contrast to the CNN-Rule and CNN+ n-gram, PPD is the high-level syntactic and semantic pattern, and it is more distinguished and has lower noise than simple linguistic rules.

V. CONCLUSION

In this paper, we design a CNN-PPD model for sentence classification. The main idea of model is to use PPD words as another input channel. The model can emphasize the important parts sentence and well generate the representations of the sentence. Then, the model benefits from the PPD mining, it representation which is always ignored in other methods. Experiments on several datasets verify that the model can learn effective features for PPD and sentence and provide enough information for judging the sentiment polarity. The case study also shows that CNN-PPD can reasonably pay attention to those words which are important to judging the sentiment polarity of sentence.

REFERENCES

[1] C. Gu, M. Wu, C. Zhang. Chinese Sentence Classification Based on Convolutional Neural Network[J]. 2017, 261(1):012008.

[2] A. Severyn, A. Moschitti. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks[C]// The, International ACM SIGIR Conference. ACM, 2015:373-382.

[3] R. Socher, E. H. Huang, J. Pennington J, A. Y. Ng, C. D. Manning. Dynamic Pooling and Unfolding Recursive Autoencoders for

Paraphrase Detection[J]. Advances in Neural Information Processing Systems, 2012, 24:801-809.

[4] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions[C]// Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John Mcintyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL. DBLP, 2011:151-161.minutes

[5] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C.D. Manning. Recursive deep models for semantic compositionality over a sentiment treebank[J]. 2013.

[6] J. L. Elman. Finding structure in time. Cognitive science , 1990, 14(2):179–211.

[7] W. Yin, H. Schütze. Multichannel Variable-Size Convolution for Sentence Classification[J]. 2016:204-214.

[8] R. Zhang, H. Lee, D. Radev. Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents[J]. 2016:1512-1521.

[9] M. Ma, L. Huang, B. Xiang, B. Zhou. Dependency-based Convolutional Neural Networks for Sentence Embedding[J]. 2015:174-179.

[10] Y. Hou, X. Zhao, D. Song, W. Li. Mining pure high-order word associations via information geometry for information retrieval[J]. Acm Transactions on Information Systems, 2013, 31(3):12.

[11] T. V. D. Cruys, T. Poibeau, A. Korhonen. Latent Vector Weighting for Word Meaning in Context[J]. Proceedings of Empirical Methods in Natural Language Processing, 2011:1012-1022.

[12] Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of EMNLP, Oct. 2014: 1746–1751.

[13] Y. Zhang, S. Roller, B. Wallace. MGNC-CNN: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification[J]. 2016.

[14] Q. V. Le, T. Mikolov. Distributed Representations of Sentences and Documents[J]. 2014, 4:II-1188.

[15] Y. Bengio, R. Ducharme, ´P. Vincent, C. Jauvin. A neural probabilistic language model. journal of machine learning research, 3(Feb):1137–1155, 2003.

[16] R. Collobert, J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM, 2008.

[17] E. H. Huang, R. Socher, C. D. Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 873– 882, 2012.

[18] T. Mikolov, M. Karafi´at, L. Burget, J. Cernock`y, S. Khudanpur. Recurrent neural network based language model. In Interspeech, pages 1045–1048, 2010.

[19] J. Pennington, R. Socher, C. D. Manning. Glove: Global vectors for word representation. In Proceedings of the EMNLP, 2014 :1532-1543.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012, 3(4):págs. 212-223.

[21] J. Duchi, E. Hazan, Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research, 12:2121–2159, 2011.

[22] B. Pang, L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, 115-124, 2005.

[23] B. Pang, L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 271-277, 2004.

[24] M. Hu, B. Liu. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 168–177, 2004.

[25] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Po.s. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the conference on empirical methods in natural language processing (EMNLP), Vol. 1631. Citeseer, 1642. 2013.

[26] N. Kalchbrenner, E. Grefenstette, P. Blunsom. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188. 2014.

[27] Z. Hu, X. Ma, Z. Liu, E. H. Hovy, E. P. Xing. Harnessing deep neural networks with logic rules. In Proceedings of ACL, 2016.

[28] S. Li, Z. Zhao, T. Liu, R. Hu, X. Du. Initializing Convolutional Filters with Semantic Features for Text Classification. Proceedings of the 2017 EMNLP, 2017 :1884-1889.

[29] K. Cho, B. V. Merrienboer B, C. Gulcehre, D. Bahdanau, F. Bougares. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.