Learning distributed sentence vectors with bi-directional 3D convolutions

Bin Liu, Liang Wang
Center of Statistical Research,
School of Statistics,
Southwestern University of Finance & Economics
Chengdu, China
{liubin@,wangliang@smail}swufe.edu.cn

Guosheng Yin
Department of Statistics
and Actuarial Science,
The University of Hong Kong
Hong Kong, China
gyin@hku.hk

Abstract

We propose to learn distributed sentence representation using the text's visual features as input. Different from the existing methods that render the words (or characters) of a sentence into images separately, we fold these images into a 3-dimensional sentence tensor. Then, multiple 3-dimensional convolutions with different lengths (the third dimension) are applied to the sentence tensor, which would act as bi-gram, tri-gram, quad-gram, and even five-gram detectors jointly. Similar to the Bi-LSTMs, these n-gram detectors learn both forward and backward distributional semantic knowledge from the sentence tensor. The proposed model uses bi-directional convolutions to learn text embedding according to the semantic order of words. The feature maps from the two directions are concatenated for final sentence embedding learning. Our model involves only a single layer of convolution which makes it easy and fast to train. We evaluate the sentence embeddings on several downstream natural language processing (NLP) tasks, which demonstrate surprisingly excellent performance of the proposed model.

1 Introduction

Mapping documents or sentences to vectors (Le and Mikolov, 2014; Pagliardini et al., 2018) is the foundation of various natural language processing (NLP) tasks, such as text classification (Kim, 2014), paraphrase detection (Socher et al., 2011), natural language inference (Bowman et al., 2015), question answering (Zhou et al., 2015), etc. The most straightforward approach to sentence representation uses the bag-of-words model that represents a sentence as a bag of its constituent words, disregarding grammar and even the word order but keeping multiplicity. Another similar approach is called Glove (Pennington et al., 2014), which takes the average of word vectors of the constituent words of a sentence. These approaches are typically efficient to train, while they ignore the sequential characteristic of the text. To account for the word order, Skip-Thought (Kiros et al., 2015) learns sentence representation in an unsupervised way inspired by the skip-gram. It aims to predict the neighboring sentences or phrases for a given sentence. However, the training process of the Skip-Thought is very slow, which motivates the FastSent (Kenter et al., 2016) to speed up the training by representing a sentence as a sum of its constituent word vectors. Although FastSent is faster than Skip-Thought in training, it sacrifices the order of words in a sentence, which is important in language models, such as the n-gram feature. For example, Gupta et al. (2019) utilize the bi-gram and even tri-gram to train their embedding model.

As discussed above, most existing works of sentence representation require pre-trained word vectors as the input or initialization. Sentence representation is taken as a downstream task of word representation. However, when human beings read a sentence or an article, their eyes in fact receive a series of text images which are then passed to the brain for recognition and understanding. Hence, a natural way of word representation is to use visual shapes of the words or characters as features directly (Shimada et al., 2016; Su and Lee, 2017; Liu et al., 2017; Sun et al., 2019; Liu and Yin, 2020). For example, Su and Lee (2017) and Shimada, Kotani, and Iyatomi (2016) take Chinese and Japanese characters as images

Proceedings of the 28th International Conference on Computational Linguistics, pages 6820–6830
Barcelona, Spain (Online), December 8-13, 2020

and apply a subsequent convolutional autoencoder to take those images as input and then output low-dimensional character embeddings. Liu and Yin (2020) extract both the forward and backward n-gram features from the text's pixel embedding.

We propose to learn the sentence embeddings using the non-pretrained word images but fold them into sentence images as input. Our model utilizes multiple bi-gram, tri-gram, quat-gram even five-gram embeddings of both forward and backward orders of words. Current research in NLP tends to use deep and complex models, which make the performance compromise to the model complexity. However, the proposed model has a lightweight structure as shown in Figure 1. In detail, we render words or characters of a sentence into images and then fold them into a 3-dimensional sentence tensor $\mathcal{X} \in \mathbb{R}^{w \times h \times l}$, where $w \times h$ is the size of the word or character image and l is the length of the sentence. Each slice $X_i \in \mathbb{R}^{w \times h}$ corresponds to a word or a character image. Furthermore, we propose to fully exploit the language feature (i.e., the word order in a sentence) with two distinctive strategies: (1) extracting the multiple n-gram features with several 3-dimensional convolutional kernels of different sizes (n is the number of words covered by the kernel); (2) learning both the forward and backward semantic information from the sentence with bi-direcitonal convolutions, as shown by Figure 1. We name the proposed model as 3D-ConvLM (3-dimensional Convolutional Language Model). We choose multiple n to integrate multiple n-gram convolutional kernels. Taking the demo in Figure 1 as an example, we use bi-gram, tri-gram, and quad-gram information together. Moreover, these n-gram information are constructed from both the normal text order and the reverse order through bi-directional convolutions. A subsequent 1-dimensional max-over-time pooling is applied to each channel of this 2-dimensional feature map output by each n-gram model with different channels. After pooling, feature maps from each n-gram model of two directions are concatenated as the output feature of the convolutional layer. Finally, three fully connected (FC) layers are used for conducting text embedding learning.

The contributions of our work are three-fold:

(1) We propose to represent a sentence or an article with a video-like 3-dimensional tensor, and each frame of this tensor represents one word in the sentence or article, which provides an alternative view to understand the NLP with computer vision techniques.

(2) We use a 3-dimensional convolutional kernel to learn the n-gram features from the text tensor.

(3) We propose to use bi-directional convolutions to extract semantic information on both the text's forward and backward orders.

The proposed 3D-ConvLM extracts and integrates multiple n-gram features during forward and backward convolutions, which further increases the flexibility of the input of text information. We evaluate 3D-ConvLM on text classification and sentence matching, and study the difference between traditional Chinese and simplified Chinese under the proposed framework.

2 Related Works

2.1 Sentence embedding

Transforming a document or a sentence into a numerical vector (i.e., embedding) according to the text's semantic meaning represents a fundamental task in downstream applications of NLP. One simple implementation of the sentence representation is to sum or average all its constituent word embeddings, such as the bag-of-words, Glove (Pennington et al., 2014), and FastSent (Kenter et al., 2016). These methods are typically efficient in training but compromise the order of words, which may cause significant information loss for text analysis.

Research works have been carried out to model the order of words when learning the distributed sentence representation (Le and Mikolov, 2014; Kiros et al., 2015; Conneau et al., 2017; Pagliardini et al., 2018; Gupta et al., 2019; Shen et al., 2019). Le and Mikolov propose Doc2vec (Le and Mikolov, 2014) to add a paragraph vector to represent the missing information from the current context. The Doc2vec is an adaptation of the Word2vec (Mikolov et al., 2013). Also inspired by the Word2vec model, the Skip-Thought (Kiros et al., 2015) learns sentence representation by predicting the neighboring sentences for any given sentences. Sent2Vec (Pagliardini et al., 2018) aims to strike a balance between matrix factorization and deep learning. Gupta et al. (2019) propose two modifications of Word2vec by considering higher-order word n-grams along with uni-gram during training. Shen et al. (2019) use InferSent (Conneau et al., 2017) for sentence embeddings based on word vectors learned by Glove (Pennington et al., 2014) or FastText (Joulin et al., 2017). Gupta et al. (2019) claim that training word embeddings along with higher n-gram embeddings helps in the removal of the contextual information from the uni-gram, resulting in better stand-alone word embeddings. All the aforementioned methods require pre-trained word vectors as input.

From a completely different perspective, the most natural way of representing text is using its visual shape, which is also how human understand the text. The pre-trained word vectors are not indispensable for sentence embedding. Intuitively, when we read an article on a screen or a book, our eyes capture the text as a series of images rather than embedding them into vectors. In other words, human understand the text with the visual information of the words, i.e., we recognize characters or words from their images that are captured by our eyes. Therefore, we believe that the pixel image, i.e., the character's morphological shape, provides the most straightforward way to represent characters and words. Motivated by this idea, several visual embedding methods (Shimada et al., 2016; Su and Lee, 2017; Sun et al., 2019) have been developed for Chinese and Japanese text understanding. However, it is very difficult to visually embed alphabetic languages such as English, because English words cannot be rendered as the same sized image as Chinese or Japanese square characters. In this work, we render English words into fixed size images. By contrast, for Chinese, each character is rendered into a squared image. Based on visual embedding, we integrate multiple n-gram embeddings of both forward and backward directions into our model.

2.2 Bi-directional models

In neural language models, both the normal order and reverse order are preferred to be used as the input. The most well-known bi-directional model is the Bi-LSTMs (Schuster and Paliwal, 1997), which accepts both the forward and backward information of text as the input. For example, Melamud et al. (2016) apply the Bi-LSTMs to a generic context embedding function from large corpora. Kawakami and Dyer (2015) represent words in the context using Bi-LSTMs and multilingual supervision.

The forward direction on the input sequence follows the order as it is and the backward applies on a reversed copy of the input sequence. The use of Bi-LSTMs may not always benefit for all sequence prediction problems, but it can improve the results in the domains where it is appropriate (Graves and Schmidhuber, 2005; Melamud et al., 2016; Kawakami and Dyer, 2015).

Substantial research works have been focused on combining the Bi-LSTMs with convolutional neural networks (CNNs). For example, the gated bi-directional CNN (Zeng et al., 2016) is a bi-directional network, which can effectively make use of multi-scale and multi-context regions of images. It is motivated by the fact that features from different resolutions and support regions can validate the existence of one another. One example is that a local rabbit ear is helpful in recognizing the rabbit from an image. However, when the local feature is that the rabbit ear is artificially located on a girl's head, it would validate the evidence to support a rabbit image. Hence, we cannot apply the gated bi-directional CNN to NLP problems. Chiu et al. (2016) propose the bi-directional LSTM-CNNs, which can automatically detect word- and character-level features using a hybrid bi-directional LSTM and CNN architectures, and thus eliminate the need for most feature engineering. In contrast, 3D-ConvLM is focused on exploiting the bi-directional semantic knowledge from the text directly.

3 Proposed Approach

3.1 Overview

Our model makes a direct connection between computer vision and NLP. We render a sentence S into a 3-dimensional tensor, where each slice of this tensor corresponds to a word (for English) or a character (for Chinese), i.e., each word from S is rendered as an image $X_i \in \mathbb{R}^{w \times h}$. We then apply 3-dimensional convolutional kernels of size $w \times h \times n$ to the "text tensor", where w and h are respectively the width and height of the character images, and n is the number of characters. In other words, the 3-dimensional

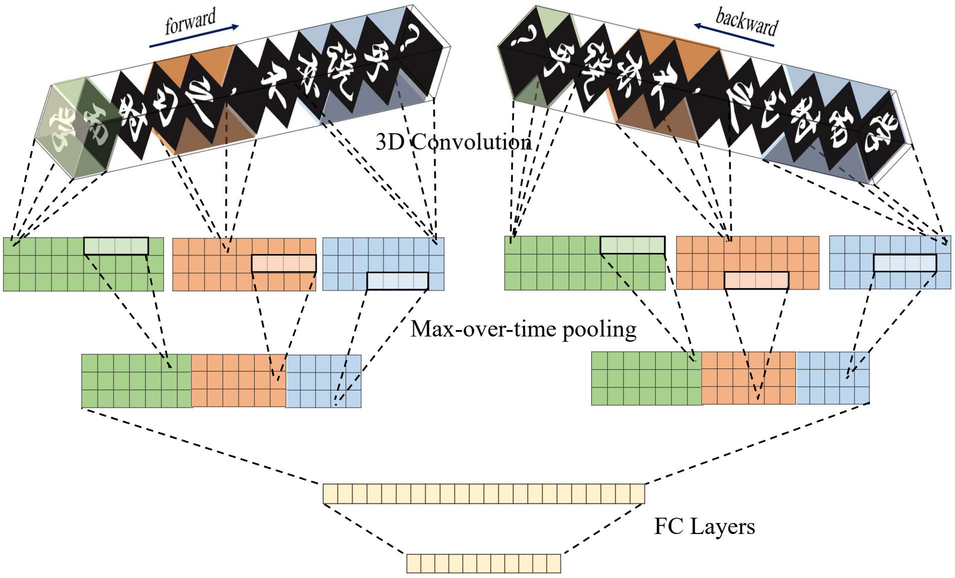


Figure 1: The Bi-directional 3-dimensional (3D) convolutional structure of the proposed model.

convolution operates n words or characters for one slide, which acts as an n-gram feature detector. The 3-dimensional convolutional kernel used here is different from the kernel in the traditional image or video processing tasks. By varying the values of n, we can obtain different n-gram detectors of different sizes. We propose to extract textual features using multiple n-gram convolutions. For example, in our experiments, n can take values of $\{2, 3, 4, 5\}$. Integration of multiple n-grams is very easy and fast to implement under the proposed framework.

In neural language modeling, textual information of the normal order and reverse order are two different inputs. For example, the Bi-LSTMs model takes both forward and backward sequences of the text as inputs. Both the forward on the input sequence as it is and the backward on a reversed copy of the input sequence are used. The integration of bi-directional information may not always benefit for all sequence prediction problems, but it can offer some improvement in those domains where it is appropriate (Graves and Schmidhuber, 2005). Traditional CNNs ignore the difference of the sequential information between the forward and backward information. In this work, we propose both the forward and backward convolutions to bridge this gap. As shown in Figure 1, three n-gram models (bi-gram, tri-gram, and quad-gram[1]) have been applied to the forward and backward text inputs. Hence, each n-gram detector from both the forward and backward models would output two 2-dimensional feature maps, in which the rows correspond to channels and the columns correspond to the n-gram features. A subsequent 1-dimensional max-over-time pooling is applied to each channel of these 2-dimensional feature maps. The max-over-time pooling means this procedure is implemented along the time dimension according to the order of a sentence, which is different from the max-over-time pooling referred in Kim et al. (2016) that takes max pooling over different convolutional kernels. After pooling, the feature maps extracted by each n-gram detector of two directions are concatenated as the output feature of the convolutional layer. Finally, three FC layers are used for conducting downstream NLP tasks, such as text classification. Figure 1 illustrates this entire process of the proposed model.

[1] In the following experiments, the five-gram model has also been used.

3.2 Discussion

CNNs have been shown to achieve excellent performance on text classification and sentiment classification (Kim, 2014). For the standard CNN on text analysis, only the forward convolution is passed to the next layer. On the contrary, we propose bi-directional convolutions to adapt our model to text data as shown in Figure 1. We add the following two more important characteristics into the CNN to make it better for text analysis: (1) We use bi-directional convolutions to extract features; and (2) we integrate multiple n-gram features as the input.

3.3 Network Implementation

The network architecture can be described in order as follows:

1. **Conv3d layer**: *kernel size = (20, 131, 3), stride = (1, 1, 1), number of kernels = 50, padding = 0*;

2. **MaxPool1d layer** (the max-over-time pooling): *kernel size = 3, stride = 3, dilation = 3, padding = 0*;

3. **FC layer 1**: *input = 1250, output = 512*;

4. **FC layer 2**: *input = 512, output = 100*;

5. **FC layer 3**: *input = 100, output = number of classes*.

The specification *stride = 3* for the **MaxPool1d** results in no overlaps in the max-over-time pooling. The kernel size of the **Conv3d layer** is (20, 131, 3). It corresponds to a tri-gram model, which is different from the bi-gram model illustrated in Figure 1.

The proposed model has a light-weight architecture, which only contains no more than $20 * 131 * 3 * 50 + 50 * (L - 6) * 1250 * 512 * 100 * num\_class$, num_class is the number of the category of each dataset. The number of parameters of 3D-ConvLM is much fewer than that of the baseline char-CNN (Zhang et al., 2015).

4 Experiments

4.1 Experimental setup

We render characters (in Chinese) or words (in English) into images of size $\mathbb{R}^{w \times h}$. Chinese characters have squared shapes, and thus can be represented by squared images with $20 \times 20 = 400$ pixels; that is, $w = h = 20$. For English words, we render each word into images, and set the height $h = 20$ and the width $w = 131$ to satisfy most of the English words with a maximum length of 17 alphabets. Our model is trained using the Adam (Kingma and Ba, 2015) with a learning rate of 0.00001.

4.2 Text classification

4.2.1 Datasets for text classification

The three datasets for text classification are introduced as follows. The THUCNews dataset[2] (Li and Sun, 2007) is generated according to the historical data obtained from the subscription channel of Sina News RSS (`https://rss.sina.com.cn/`) from year 2005 to 2011. The Toutiao news dataset collects the text from the Toutiao App[3], and each item contains the title and keywords of the news. We concatenate the title and keywords as one sample where samples of lengths shorter than 5 are removed and thus the remaining 380,455 samples are used for training and testing (`https://github.com/fate233/toutiao-text-classfication-dataset`). There are 382,688 item documents in the raw data. We filter out sentences with lengths outside of the range $[5, 100]$ and, as a result, there remain 380,455 item documents, i.e., 99.41% sentences with their lengths falling in the range of $[5, 100]$. The Dianping dataset consists of user reviews from online restaurants (`http://www.dianping.`

[2] http://thuctc.thunlp.org

[3] It is also called Jinri Toutiao, which is a news and information content platform of the company ByteDance. It is one of the largest mobile platforms of content creation, aggregation and distribution in China. `https://www.toutiao.com/`

com/) (Zhang and LeCun, 2017), which contains 2,500,000 samples. Each sample is a review with a score ranging from 1 star to 5 stars. We mark a review as a positive sentiment if the star value equals 4 or 5, and negative otherwise.

For the three datasets, the sample sizes for training, validation, and testing are given in Table 1.

4.2.2 Baselines

For comparison, we consider four baseline methods described as follows:

- The character-level convolutional neural networks (char-CNN) (Zhang et al., 2015);
- CNN for text classification on top of the distributed word vectors obtained via Word2vec (Kim, 2014);
- CNN for text classification on top of the one-hot word vectors;
- FastText (Joulin et al., 2017).

We experiment with three variants of 3D-ConvLM:

- 3D-ConvLM using both the bi-directional convolution and multiple n-grams inputs, as shown in Figure 1.
- 3D-ConvLM using the bi-directional convolution and 3-gram detector with the filter size $w \times h \times 3$.
- 3D-ConvLM using only the 3-gram which has two convolutional layers and the filter size is $w \times h \times 3$.

Table 1: Split of the sample size for training, validation and testing of the datasets for text classification.

Datasets	Training	Validation	Testing	Classes	Average length	Content
THUCNews	50,000	5,000	10,000	10	251	News
Toutiao	266,318	37,666	76,471	15	38	Title and keywords
Dianping	1,750,000	250,000	500,000	2	148	Food reviews

4.2.3 Results on text classification

Testing results on the accuracy of our models in comparison with other methods are shown in Table 2. The proposed model with bi-directional convolutions and $\{2, 3, 4, 5\}$-gram achieves superior performances on the text classification compared with the others. It is worth emphasizing that both the bi-directional convolution and multiple n-gram detector contribute to the final performance, which can be supported by the experimental results of the first three rows of Table 2. By comparing the results in row 2 and row 3, we observe that the model with bi-directional convolution in row 2 indicates better accuracy than the model in row 3. Results in row 1 and row 2 in Table 2 also suggest that integrating multiple n-gram improves the performance.

4.3 Interpretation of 3-dimensional convolution

The 3-dimensional convolutional kernel acts as an n-gram detector in our model. As shown in Figure 1, the conventional kernel operates on n frames (i.e., n words) at a time, which thus corresponds to an n-gram detector. For a sentence S of length l, we can generate a feature vector $\mathbf{u} \in \mathbb{R}^{l-1}$, which is a continuous n-gram feature of S. By applying k different kernels, we obtain a feature map $\mathbf{U} \in \mathbb{R}^{k\times(l-1)}$ for each n-gram detector.

When carrying out the testing, we also render a test sentence $S = \{v_1, v_2, \ldots, v_l\}$ into a text video $\mathcal{X} = \{X_1, X_2, \ldots, X_l\}$. We input the text video $\mathcal{X}$, which outputs a corresponding feature map $\mathbf{U}$.

Table 2: Comparisons of the accuracy between our three pixel embedding models, 3D-ConvLM, and four baseline methods under three datasets.

Methods	THUCNews	Toutiao	Dianping
3D-ConvLM (bi-directional + $\{2, 3, 4, 5\}$-gram)	**0.94**	**0.86**	**0.77**
3D-ConvLM (bi-directional + 3-gram)	0.92	0.85	0.76
3D-ConvLM (3-gram)	0.89	0.84	0.75
char-CNN	0.92	0.84	0.76
CNN one-hot	0.86	0.78	0.69
CNN Word2vec	0.91	0.81	0.68
FastText	0.91	0.79	0.73

Its element $\mathbf{U}_{i,j}$ corresponds to the convolutional result between the i-th 3-dimensional convolutional kernel and the j-th n-gram. A larger value of $\mathbf{U}_{i,j}$ indicates that the j-th n-gram of the input sentence S is more relevant for the classification task (selected by kernel i). By identifying the maximum $\widetilde{\mathbf{U}}_{i,j}$ of all elements in $\mathbf{U}$, we can easily find out the most relevant n-gram for the task of classification within the sentence S, where $j + 1 \leq l$, and l is the number of words in S.

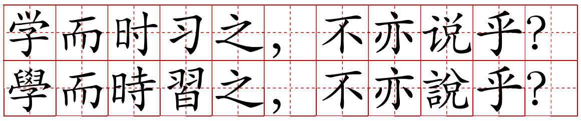
We visualize the weighted bi-grams according to the first layer of the network trained on the task of topic classification of the dataset THUCnews. It has ten classes as shown in Table 3. There are 10,000 testing samples for all categories. Table 3 illustrates the top five bi-grams associated with each category.

Almost all the top five bi-grams detected by the proposed model (bi-directional + $\{2, 3, 4, 5\}$-grams) are relevant to the corresponding topics. Taking the topic "Finance" as an example, the detected bi-grams, "基金" in Chinese and "fund" in English, are strongly matched with its semantic topic. We mark it in blue. We also see that the third bi-grams are "记者" in Chinese and "journalist" in English, which may or may not have any relationship with finance so that we color it in orange. The fifth most relevant bi-grams of "Finance" suggested by 3D-ConvLM is "可能", which however is not relevant and we color it in red.

Because 3D-ConvLM does not need to conduct preprocessing, such as segmentation (for Chinese), some special bi-grams without any semantic meaning could be detected, for instance, "图) ", "》 报", "月2", and "》 新". Although these bi-grams are meaningless, they do exist in the corpus with high frequency.

4.4 Simplified Chinese versus traditional Chinese

Chinese characters are possibly the oldest continuously used but most complex systems of writing in all languages. There are two coexisting writing systems, i.e., simplified Chinese and traditional Chinese. The two writing systems are used to write almost all Chinese dialects[4]. As shown in Figure 2, the upper row illustrates characters of simplified Chinese, and the lower row displays the same characters but in traditional Chinese, which clearly have more strokes than the simplified Chinese. Over the years, there have been extensive debates about traditional and simplified Chinese. For example, what are the differences between traditional and simplified Chinese? And which one is more efficient?



学而时习之，不亦说乎?

學而時習之，不亦說乎?

Figure 2: An example of simplified Chinese versus traditional Chinese.

In this section, we compare the differences between the simplified Chinese and traditional Chinese under the framework of 3D-ConvLM. We render the three datasets into both simplified Chinese and traditional Chinese, then run the text classification tasks on them for 10 times. From the results in Table

[4] https://unitedlanguagegroup.com/blog/traditional-chinese-vs-simplified-chinese-whats-the-difference/

Table 3: Top five bi-grams of 10 different topics from the Chinese dataset THUCNews by the proposed model: bi-directional + $\{2, 3, 4, 5\}$-gram. Each one is listed in the format of [bi-gram in Chinese]+(its English translation)+(frequency). We use three different colors to indicate the relevancy between the topic and bi-grams: blue means strongly related, orange means possibly relevant, and red means irrelevant.

Topic	Top five bi-grams for 10 different topics of THUCNews based on frequency.
Finance	基金 (Fund) (12203), 投资 (Invest) (484), 记者 (Journalist) (182), 公司 (Company) (163), 可能 (Possible) (130)
House	图) (Drawing) (4292), 新浪 (Sina Co., ltd) (2131), 投资 (Invest) (815), 开盘 (Open house) (694), 地产 (Real estate) (659)
Stock	投资 (Invest) (4696), 公司 (Company) (1540), 新浪 (Sina Co., ltd) (1400), 美国 (U.S.A.) (1237), 可能 (Possible) (877)
Education	高考 (University entrance examination) (4534), 学生 (Student) (3031), 图) (Drawing) (2092), 考生 (Examinee) (960), 留学 (Study aboard) (241)
Technology	图) (Drawing) (2159), 图片 (Picture) (1897), 影像(Image) (598), 新浪 (Sina Co., ltd) (163), 公司 (Company) (388)
Society	男子 (Man) (3011), 法院 (Court) (2766), 学生 (Student) (1167), 报讯 (News from newspaper) (850), 通讯 (Communication) (528)
Politics	月2 (Month 2) (2101), 》报 (Newspaper) (760), 法院 (Court) (641), 美国 (U.S.A.) (559), 图) (Drawing) (516)
Sports	新浪 (Sina Co., ltd) (3622), 球队 (National team) (1573), 足球 (Soccer) (874), 比赛 (Game) (663), 篮球 (Basketball) (595)
E-game	游戏 (E-game) (7630), 》新 (New) (813), 娱乐 (Entertainment) (471), 月2 (Month 2) (141), 战队 (Clans) (104)
Entertainment	娱乐 (Entertainment) (2587), 新浪 (Sina Co., ltd) (1098), 报讯 (News from newspaper) (971), 媒体 (Media) (841), 月2 (Month 2) (411)

4, we can see that although simplified Chinese and traditional Chinese characters have different forms, they deliver comparable capability on language expression.

4.5 Sentence matching

In this section, we evaluate the relatedness and entailment relation between two sentences. The relatedness and entailment relation are defined based on a sentence pair $(S_{\mathbf{A}}, S_{\mathbf{B}})$. The relatedness is a 5-point score that quantifies the degree of semantic relatedness between sentences; the entailment relation between $S_{\mathbf{A}}$ and $S_{\mathbf{B}}$ could be: entailment, contradiction, and neutral. To adapt the proposed 3D-ConvLM to these two tasks, we modify it by inputting sentence $(S_{\mathbf{A}}, S_{\mathbf{B}})$ separately. Then multiple n-gram detectors are applied to $S_{\mathbf{A}}$ and $S_{\mathbf{B}}$ to output two separate feature maps.

1. Split and render the sentence pair $(S_{\mathbf{A}}, S_{\mathbf{B}})$ into 3-dimensional tensors $\mathcal{X}_{\mathbf{A}}$, $\mathcal{X}_{\mathbf{B}}$;
2. Apply multiple bi-directional n-gram detectors to $\mathcal{X}_{\mathbf{A}}$, $\mathcal{X}_{\mathbf{B}}$ separately and output feature maps $\mathbf{A}$, $\mathbf{B}$;
3. Impose 1-dimensional max-over-time pooling on $\mathbf{A}$, $\mathbf{B}$ and output $\mathbf{A}'$, $\mathbf{B}'$ respectively;
4. Calculate the angle and distance $|\mathbf{A}' - \mathbf{B}'|$, $|\mathbf{A}' \circ \mathbf{B}'|$ between $\mathbf{A}'$ and $\mathbf{B}'$ (Tai et al., 2015) and concatenate them into $(|\mathbf{A}' - \mathbf{B}'|, |\mathbf{A}' \circ \mathbf{B}'|)$, where $|\cdot|$ is an absolute element-wise difference, $\circ$ is the Hadamard product;
5. Feed the FC layers $(|\mathbf{A}' - \mathbf{B}'|, |\mathbf{A}' \circ \mathbf{B}'|)$ for relatedness and entailment prediction.

Table 4: Comparisons of the accuracy of 10 tests between simplified Chinese and traditional Chinese. We render the three datasets into both simplified Chinese and traditional Chinese and then conduct tests for 10 times.

Datasets	Chinese	Bi-directional + $\{2, 3, 4, 5\}$-gram									
Toutiao	Simplified	0.86	0.85	0.86	0.85	0.86	0.86	0.86	0.85	0.86	0.86
	Traditional	0.85	0.86	0.85	0.85	0.85	0.85	0.85	0.86	0.85	0.86
THUcnews	Simplified	0.94	0.94	0.94	0.94	0.95	0.94	0.94	0.95	0.94	0.94
	Traditional	0.94	0.95	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
Dianping	Simplified	0.74	0.74	0.73	0.74	0.75	0.74	0.74	0.74	0.74	0.74
	Traditional	0.74	0.74	0.74	0.74	0.73	0.73	0.74	0.74	0.74	0.73

4.5.1 Datasets for sentence matching

The SICK dataset (Marelli et al., 2014) contains about 10,000 English sentence pairs. Each pair was annotated for relatedness (SICK-R) and entailment (SICK-E) by means of crowdsourcing[5]. Samples in STS14 (Agirre et al., 2014) are also labeled as well as the SICK-R that contains 36000 sentence pairs.

4.5.2 Baselines

We run the following baselines with the SentEval (Conneau and Kiela, 2018), which is a sentence embeddings evaluation toolkit[6].

- FastText (Joulin et al., 2017) with the bag-of-words (BoW).
- Glove (Pennington et al., 2014) with the bag-of-words (BoW).
- Skip-Thought (Kiros et al., 2015).

Table 5: Results on the testing set of three tasks. We evaluate the SICK-E with classification accuracy, SICK-R, and STS14 with both Pearson/Spearman correlations.

Methods	SICK-E	SICK-R	STS14
3D-ConvLM	0.79	0.73/0.7	**0.61/0.63**
FastText+BoW	0.78	0.73/0.69	0.54/0.56
Glove+BoW	0.78	**0.79/0.71**	0.54/0.55
Skip-Thought	**0.82**	0.59/0.62	0.29/0.35

4.5.3 Results of sentence matching

The results of sentence matching are shown in Table 5. For SICK-E and SICK-R, we see that the proposed model achieves comparable results with the FastText+BoW, while Skip-Thought and Glove+BoW perform as the best two. But for the STS14 set, our model achieves the highest Pearson/Spearman correlations. Although the proposed model does not present a remarkable performance on all the tasks, it has a light-weight structure for training.

5 Conclusion

Visual embedding of the text has been studied extensively in recent years. CNNs can deliver a competitive performance in comparison with LSTM on text data analysis. We propose a bi-directional CNN to learn text embedding according to the semantic order of sentences. In our model, visual signals of each character can be extracted by multiple n-grams in both the normal order and reversed order. We conduct text classification and sentence matching on several datasets to evaluate the performance of our model. Within the proposed framework, we also study the difference between the simplified Chinese and traditional Chinese.

[5] https://zenodo.org/record/2787612#.XeZNsPl3iUk

[6] https://github.com/facebookresearch/SentEval

Acknowledgement

The research was supported by the Fundamental Research Funds for the Central Universities, NO. JBK1806002, and the Research Grants Council of Hong Kong, NO. 17307218.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Prakhar Gupta, Matteo Pagliardini, and Martin Jaggi. 2019. Better word embeddings by disentangling contextual n-gram information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 933–939.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 427–431.

Kazuya Kawakami and Chris Dyer. 2015. Learning to represent words in context with multilingual supervision. *arXiv preprint arXiv:1511.04623*.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese cbow: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 941–951.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI Conference on Artificial Intelligence*, pages 2741–2749.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196.

Jingyang Li and Maosong Sun. 2007. Scalable term selection for text categorization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 774–782.

Bin Liu and Guosheng Yin. 2020. Chinese document classification with bi-directional convolutional language model. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1785–1788.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning character-level compositionality with visual features. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2059–2068.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Dinghan Shen, Pengyu Cheng, Dhanasekar Sundararaman, Xinyuan Zhang, Qian Yang, Meng Tang, Asli Celikyilmaz, and Lawrence Carin. 2019. Learning compressed sentence representations for on-device text processing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 107–116.

Daiki Shimada, Ryunosuke Kotani, and Hitoshi Iyatomi. 2016. Document classification through image-based character embedding and wildcard training. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3922–3927. IEEE.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances In Neural Information Processing Systems*, pages 801–809.

Tzuray Su and Hungyi Lee. 2017. Learning chinese word representations from glyphs of characters. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 264–273.

Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. Vcwe: Visual character-enhanced word embeddings. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2710–2719.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566.

Xingyu Zeng, Wanli Ouyang, Bin Yang, Junjie Yan, and Xiaogang Wang. 2016. Gated bi-directional CNN for object detection. In *European Conference on Computer Vision*, pages 354–369.

Xiang Zhang and Yann LeCun. 2017. Which encoding is the best for text classification in chinese, english, japanese and korean? *arXiv preprint arXiv:1708.02657*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2015. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*.