EXTRACTOR-ATTENTION NETWORK: A NEW ATTENTION NETWORK WITH HYBRID ENCODERS FOR CHINESE TEXT CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Chinese text classification has received more and more attention today. However, the problem of Chinese text representation still hinders the improvement of Chinese text classification, especially the polyphone and the homophone in social media. To cope with it effectively, we propose a new structure, the Extractor, based on attention mechanisms and design novel attention networks named Extractor-attention network (EAN). Unlike most of previous works, EAN uses a combination of a word encoder and a Pinyin character encoder instead of a single encoder. It improves the capability of Chinese text representation. Moreover, compared with the hybrid encoder methods, EAN has more complex combination architecture and more reducing parameters structures. Thus, EAN can take advantage of a large amount of information that comes from multi-inputs and alleviates efficiency issues. The proposed model achieves the state of the art results on 5 large datasets for Chinese text classification.

1 INTRODUCTION

Recently, Chinese text classification, as an important task of Chinese natural language processing (NLP), is extensively applied in many fields. Deep learning has gotten great results on Chinese text classification. However, the relevant studies are still insufficient compared with English, especially the method of Chinese text representation or encoding. It is considered to be closely related to the result of Chinese text classification models. Specifically, there are some issues in previous representation methods: (i) The word embedding (Le & Mikolov (2014); Mikolov et al. (2013); Pennington et al. (2014)) is the most common method to represent the text, but it may become less effective when processing texts with the ambiguous word boundary such as Chinese texts. (ii) The character embedding (Zhang et al. (2015)) can avoid the word segment. However, using Pinyin characters loses the ideographic ability of Chinese characters, and using Chinese characters requires more training data because there are thousands of Chinese characters that are often used in daily life. (iii) Both the word embedding and the Chinese character embedding are hard to encode some intricate Chinese language phenomena about pronunciations, such as the polyphone and the homophone.

We notice that humans have associated the word or character with the corresponding pronunciation and remembered them in the process of learning the language. Thus, when humans read texts in daily life, they spontaneously associate with the corresponding voices. It is very difficult for computers and usually ignored by traditional text classification method. Moreover, using the voice can cope with some representation issues of Chinese characters or words better, The polyphone and the homophone are 2 typical examples. The former means different pronunciations and meanings are from the same character, and the latter means the same pronunciations are from different characters, which are usually used to represent similar meanings in social media. And inspired by recent multimedia domain methods (Gu et al. (2018)), the extra audio information can obtain better results. However, large amounts of corresponding audio data are required difficultly. Pinyin can precisely express the pronunciation by no more than 6 letters and is easily generated from texts, and it also solves representation issues of Chinese characters or words.

There are some typical examples that illustrate these points in detail. Table 1 shows an example (sentence1) of the homophone of social medias.There is a homophone ”鸭梨山大” , the pronunciation

Table 1: Examples of social media. The Bold is the polyphone (including Chinese characters and Pinyin).

sentence1:	我只能说东西是好东西，1号的订单，6号才到，定单是两件，一件韵达，一件中通，韵达2天到，中通6天到，商家玩分开送，真是让买家**鸭梨山大(压力山大)**！ wǒ zhǐ néng shuō dōng xī shì hǎo dōng xī，1 hào de dìng dān，6 hào cái dào，dìng dān shì liǎng jiàn，yī jiàn yùn dá，yī jiàn zhōng tōng，yùn dá 2 tiān dào，zhōng tōng 6 tiān dào，shāng jiā wán fēn kāi sòng，zhēn shì ràng mǎi jiā **yā lí shān dà**！
sentence2:	大学英语六级考试：优选真题 标准模拟 没有王长喜**好**用，后悔了 dà xué yīng yǔ liù jí kǎo shì：yōu xuǎn zhēn tí biāo zhǔn mó nǐ méi yǒu wáng zhǎng xǐ **hǎo** yòng，hòu huǐ le
sentence3:	有一点点小(我个人的喜**好**)，勉强吧 yǒu yī diǎn diǎn xiǎo (wǒ gè rén de xǐ **hào**)，miǎn qiǎng ba

(Pinyin) and the meaning of which are the same as ”压力山大”. Table 1 also shows some examples (sentence2 and sentence3) of the polyphone of social medias. The pronunciation (Pinyin) and the meaning of ”好” are different in two sentences. Besides,”hào” can represent ”好” in sentence3 or ”号” in sentence1. In fact, it can represent the pronunciation of dozens of Chinese characters.

By those examples, we foucs on some points: In Chinese texts, some intricate language phenomena about pronunciations relatively easier to be recognized by a simple Pinyin encoder than by a complex Chinese character or word encoder. And most of language phenomena about glyph are the opposite. Based on the above points, we propose a new hybrid encoder (including word encoder and Pinyin character encoder) network to obtain better results for Chinese text classification, we call it Extractor-attention network (EAN). Inspired by Transformer (Vaswani et al. (2017)), we also propose a new structure named the Extractor. The Extractor includes a multi-head self-attention mechanism with separable convolution layers (Chollet (2017)). In EAN, the Extractor is used to encode the information of Pinyin. Besides, it is repeatedly used to combine word encoder with Pinyin encoder. Compared with previous hybrid encoder methods, our method has relatively simple encoders and a complicated combination part, which uses a deep self-attention mechanism. It makes EAN assign weights between features extracting by each encoder more accurately and avoid huge feature maps. Moreover, we use pooling layers for downsampling and separable convolution layers to compress parameters. Therefore, the Extractor network represent the Chinese text well, improve the classification accuracy, and the computational cost is relatively cheap. The experimental results show that our model outperforms all baseline models on all datasets, and has fewer parameters in comparison to similar works.

Our primary contributions (i) Inspired by human language learning and reading, we design a novel method to solve the text representation issue of Chinese text classification, especially the language phenomena about pronunciations such as the polyphone and the homophone. To the best of our knowledge, this is the first time that a hybrid encoding method including Pinyin has been used to solve those language phenomena expression problem. (ii) We propose a new attention architecture named the Extractor to experss Chinese texts information. Besides, to better represent Chinese texts, we design a new hybird encoder method EAN based on the Extractor. We also propose a complex attention method to combine word encoder with Pinyin encoder effectively, which can commendably balance the amount of information transmitted by 2 encoders. (iii) Our method is able to surpass previous methods. It can get the state of the art results on public datasets.

2 Related Work

Today deep neural networks have been widely used in text classification. Compared with traditional methods Pang et al. (2002), these methods do not rely on hand-crafted features. Deep learning usually represents or encodes texts as feature vectors and classifies them. The first step of text representation is to convert texts to low dimension vectors. The embedding methods are often utilized in this process. These methods include pre-trained word embedding (Le & Mikolov (2014); Mikolov et al. (2013); Pennington et al. (2014)), character embedding (Conneau et al. (2017); Zhang et al. (2015)), and word embedding without pre-trained (Blunsom et al. (2014)). After embedding, most of the deep neural networks use the convolutional neural network (CNN) or the recurrent neural network (RNN) and RNN variants. CNN structures can effectively screen out the location information from the text by the convolutional layer and the pooling layer. RNN structures and variants, especially Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber (1997)), can obtain good results in capturing sequence features. Both CNN-based methods (Conneau et al. (2017); Blunsom et al. (2014); Kim (2014); Kim et al. (2016); Zhang et al. (2015)) and RNN-based methods (Tang et al. (2015)) have achieved outstanding accomplishments in text classification. Besides, the attention mechanisms are usually used in NLP to capture relatively more critical features. Some of them are based on RNN (Gu et al. (2018); Yang et al. (2016)) or CNN (Gu et al. (2018)). Wang et al. (2018) use CNN within the attention mechanism. Others are based solely on attention mechanisms such as Vaswani et al. (2017), which performs exceptionally well in many tasks of NLP. It means that using attention mechanisms entirely without CNN or RNN to represent texts is perfectly feasible. At last, there is a softmax (multiclass classification) or sigmoid (binary or multi-label classification) classifier. Sometimes full-connection layers may be added in front of it such as Zhang & LeCun (2017).

Compared with the mainstream English text classification, the most significant difference of Chinese text classification is the text presentation approach. Sometimes they are not different except for embedding (Conneau et al. (2017); Li et al. (2018); Zhang et al. (2015)). Moreover, Shi et al. (2015) propose the Radical embedding which is similar to Mikolov et al. (2013) but uses Chinese radicals instead of words.Zhuang et al. (2017; 2018) utilize Chinese character strokes and multi-layers CNN to represent Chinese text. Liu et al. (2017) propose the visual character embedding that creates an image for the characters and employs CNN to process the image. The experiments show that it performs well in different languages data including Chinese, Japanese, and Korean.

Chinese Pinyin has gained popularity among relative researchers in recent years. It is used in character embedding (Conneau et al. (2017); Zhang et al. (2015)) at the primary stage. Then Pinyin is regarded as the Chinese word in pre-trained word embedding methods or is combined with the Chinese word as training data (sometimes also including the Chinese character). Zhang & LeCun (2017) propose a variety of encoder methods of Chinese, Japanese, Korean and English. These methods mainly consist of differently simple encoding of character, word, and romanization word. Liu et al. (2018) propose a multi-channel CNN for Chinese sentiment analysis. The channels include word, character, and Pinyin. This model shows that the combination always performs better than using the Chinese word or Pinyin alone.

Those Chinese text classification methods have gotten good results in different datasets. However, there are still some disadvantages: Some methods (Liu et al. (2017); Zhang & LeCun (2017)) are relatively straightforward so that not do well in lengthy and complicated text data, and some methods (Conneau et al. (2017)) have quite a few parameters result in relatively inefficiency.

3 Extractor-Attention Network (EAN)

Our Extractor-based method can be divided into several parts: the word encoder, the Pinyin encoder, the combination part, and a classifier. We illustrate these parts in the following sections. Besides, the attention mechanism is also repeatedly employed inside and outside of the Extractor, and thus we call this method the Extractor-attention network (EAN). The method architecture is shown in Figure 1. In EAN, Batch Normalization (BN) (Ioffe & Szegedy (2015)) is used after all convolutional layers. The activation function is the rectified linear unit (RELU) for all convolutional layers and full-connected layers.

3.1 Word Encoder

In the embedding part,the pre-trained word embedding method is employed like most text classification methods. There are 3 consecutive operations after it: Gaussian noise, dropout, and BN. They preclude overfitting or making the model converged faster. A single separable CNN (Chollet (2017)) layer is placed at the end.

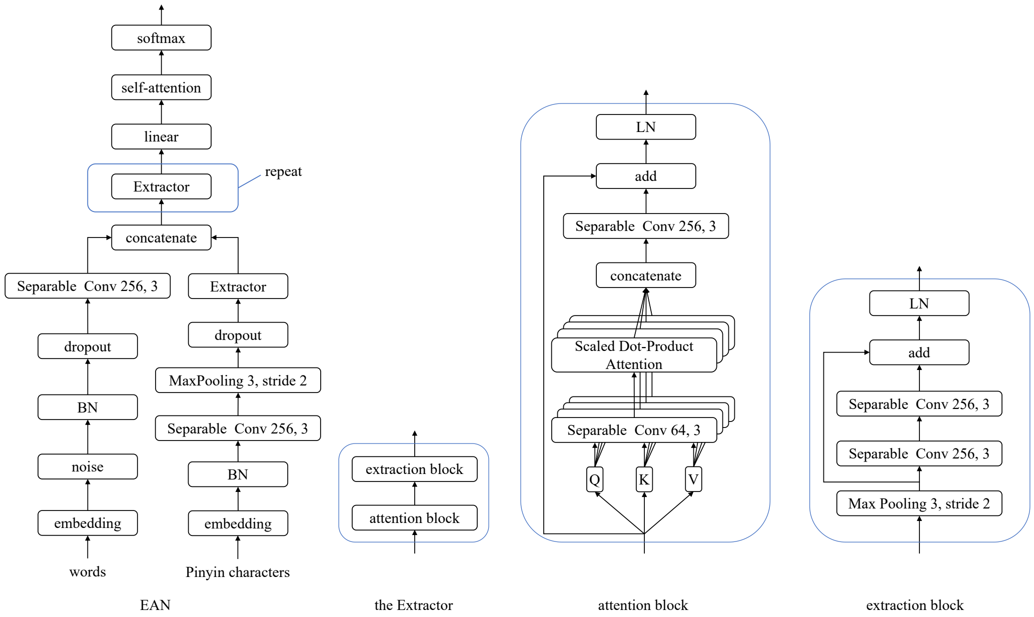

Figure 1: EAN and the Extractor architecture. BN indicates Batch Normalization (Ioffe & Szegedy (2015)). LN indicates Layer Normalization (Ba et al. (2016)).

3.2 Pinyin Encoder

The Pinyin encoder, which consists of 2 parts: the character embedding part and an Extractor, is designed to represent audio information from Chinese texts data and avoid the issue of word segment accuracy. It can supply some information which is difficult to be extracted from Chinese texts, especially the polyphone and the homophone.

In Pinyin character embedding part, the embedding layer is similar to Zhang & LeCun (2017). The characters consist of Pinyin letters, digits and punctuations. We use a Gaussian distribution to initialize the embedding weights. Therefore, the Gaussian noise operation is not used. After the embedding layer, we employ BN, a combination of separable CNN and max pooling, and dropout.

The Extractor is composed of an attention block and an extraction block, as roughly shown in Figure 1. The residual connection (He et al. (2016)) is employed in each block, which can alleviate gradient issues, speed up training, and strengthen feature propagation. Layer Normalization (LN) (Ba et al. (2016)) is applied after the residual connection.

The attention block The attention block extracts features by assigning weights to itself. Some attention structures that include self-attention and multi-head attention have gotten great results in many NLP tasks, especially the Transformer (Vaswani et al. (2017)). Thus, nonlinear multi-head self-attention structure is employed in the attention block to enhance the representation ability of the model. The original linear operation of multi-head attention is replaced by the separable CNN. Compared with linear operations such as fully-connection layers, CNN is more capable of capturing local and position-invariance features. Besides, CNN is a faster computation due to parallel-processing friendly, peculiarly separable CNN with fewer parameters. These properties are required for Chinese text representation and classification.

The input of this block is the output of the Pinyin character embedding part. Define $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ as the matrixes which consist of queries, keys, and values, respectively. In self-attention, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are the identical matrixes of size $l \times d$, where l is the input length, d is the number of the input channels. To obtain different attention functions, different representation subspaces should be generalized. In order to achieve it, we have:

$$\mathbf{Q}_s = [\mathbf{Q}_{:,:,1}, \mathbf{Q}_{:,:,2}, \dots, \mathbf{Q}_{:,:,n}] \tag{1}$$

$$\text{where} \quad \mathbf{Q}_{:,:,i} = \text{Separable Conv1D}(\boldsymbol{Q}) \tag{2}$$

$$\mathbf{K}_s = [\mathbf{K}_{:,:,1}, \mathbf{K}_{:,:,2}, \dots, \mathbf{K}_{:,:,n}] \tag{3}$$

$$\text{where} \quad \mathbf{K}_{:,:,i} = \text{Separable Conv1D}(\boldsymbol{K}) \tag{4}$$

$$\mathbf{V}_s = [\mathbf{V}_{:,:,1}, \mathbf{V}_{:,:,2}, \dots, \mathbf{V}_{:,:,n}] \tag{5}$$

$$\text{where} \quad \mathbf{V}_{:,:,i} = \text{Separable Conv1D}(\boldsymbol{V}) \tag{6}$$

Where n is the number of heads, Separable Conv1D is the separable 1D convolution function, and $\mathbf{Q}_{:,:,i}, \mathbf{K}_{:,:,i}, \mathbf{V}_{:,:,i} \in \mathbb{R}^{l \times d_k}$ are the i-th matrix of $\mathbf{Q}_s, \mathbf{K}_s, \mathbf{V}_s$, respectively. Define d_k is the number of channels of $\mathbf{Q}_{:,:,i}, \mathbf{K}_{:,:,i}, \mathbf{V}_{:,:,i}$:

$$d_k = \frac{d}{n} \tag{7}$$

For each head $\boldsymbol{H}_i$, the Scaled Dot-Product Attention is employed to capture the internal relationship:

$$\boldsymbol{H}{:}, i = \text{softmax}(\frac{\mathbf{Q}_{:,:,i} \cdot \mathbf{K}^T_{:,:,i}}{\sqrt{d_k}})\mathbf{V}_{:,:,i} \tag{8}$$

All the heads are concatenated, then processed by a separable CNN layer. Define $\boldsymbol{P}$ as the output of the CNN, and it is also the output of block:

$$\boldsymbol{P} = \text{Separable Conv1D}(\text{Concatenate}(\boldsymbol{H}{:}, 1, \boldsymbol{H}{:}, 2, \dots, \boldsymbol{H}{:}, n)) \tag{9}$$

The extraction block The extraction block compresses the feature maps and further extracts features. Compared with the word embedding, there is no word boundary issue in the Pinyin character embedding. However, the Pinyin character embedding requires a much longer length than the word embedding. Thus, the feature maps of Pinyin encoder may be too large to be processed efficiently. The filtration of feature maps is employed to alleviate this problem, which is why the extraction block is designed. At first, a downsampling operation by max pooling is used to primarily reduce feature maps of the output of the attention block. To further extract the relative spatial information and introduce more nonlinear transformation, 2-layers separable CNN is used after the max pooling layer. By this block, the feature maps become narrow.

3.3 Combination Part and Classifier

The key problem of the hybrid encoder method lies in combining the encoders.Traditional combination methods often use the simple features concatenation (Liu et al. (2018)) or the complicated encoders (including attention structures) with straightforward features combination (Amplayo et al. (2018)). We choose relatively more uncomplicated encoders and more complex combination ways to avoid redundancy and overmuch parameters. The combination part concatenates the output of word and Pinyin encoder at the first step. And then the Extractor is employed repeatedly to extract long-term dependencies and global features. Besides, Extractors can effectively reduce feature maps. The Extractor structure is similar to that of Pinyin encoder. Finally, a Scaled Dot-Product Attention is employed to weight the output of the final Extractor by the self-attention scores. We do not choose the global max pooling layer or the flatten layer, because the global max pooling layer is coarse, and the flatten layer has too many parameters. Define $\boldsymbol{X} \in \mathbb{R}^{d_l \times d_x}$ is the output matrix of the final Extractor. d_l is the input length of $\boldsymbol{X}$, d_x is the number of channels of $\boldsymbol{X}$.The self-attention scores $\boldsymbol{A}$ can be computed by $\boldsymbol{X}$:

$$\boldsymbol{A} = \text{softmax}(\frac{\boldsymbol{X}\boldsymbol{X}^T}{\sqrt{d_x}}) \tag{10}$$

The final hybrid representation $\boldsymbol{f}$ is the sum of weighted features by $\boldsymbol{A}$:

$$\boldsymbol{f} = \sum \boldsymbol{A}{:}, i \cdot \boldsymbol{X}{:}, i \tag{11}$$

Where $\boldsymbol{A}{:}, i \in \boldsymbol{A}$, $\boldsymbol{X}{:}, i \in \boldsymbol{X}$. $\boldsymbol{f}$ is the output of the combination part, and is also the input of the classifier. The classifier is the final part, which consists of 1 or 3 full-connected layers and a softmax layer. The dropout and BN are used after each full-connected layer in this part.

Table 2: Datasets information, including the number of classes, number of training samples, and number of testing samples.

Dataset	Classes	Train	Test
Dianping	2	2000 K	500 K
JD.b	2	4000 K	360 K
JD.f	5	3000 K	250 K
Ifeng	5	800 K	50 K
Chinanews	7	1400 K	112 K

4 EXPERIMENTS

Benchmark Datasets We experiment EAN on 5 benchmark datasets[1] of text classification proposed by Zhang & LeCun (2017). Specifically, Ifeng and Chinanews are news topic classification datasets. Dianping, JD.b, and JD.f are sentiment classification datasets on user review. All datasets are Chinese text datasets. The summary statistics of datasets are shown in Table 2. We selected 10K documents from the training data for use as the validation set on each dataset.

Baselines We compared EAN with various methods, including EmbedNet, GlyphNet, OnehotNet, Linear model (multinomial logistic regression), fastText (Joulin et al. (2017)), and the EAN without the hybrid encoder (removing the concatenation layer and Pinyin encoder). All experiments data of those baseline methods come from Zhang & LeCun (2017). We will omit an exhaustive background description of the baseline methods and refer readers to Zhang & LeCun (2017). Besides, to comfortably compare the parameters between EAN and other methods, especially the hybrid methods, we design a comparison baseline based on EAN. All the Extractor are replaced by the Transformer encoder structure (Multi-Head Attention and Feed-Forward Networks). Thus, we name it TAN. We tested TAN with hybrid encoder and TAN with word encoder. We also compared the parameters with some text classification model including Bi-BloSAN (Shen et al. (2018)) (the result are cited from Yu et al. (2017)) and VDCNN (Conneau et al. (2017)).

Setup In word embedding, we employed Jieba, a word segmentation package, to process Chinese texts and used the SGNS vectors (Wikipedia-zh (Word + Ngram)) by Li et al. (2018) as the embedding initialization. In Pinyin character embedding, we obtained the Pinyin texts by the pypinyin package. The character embedding weights were initialized from a Gaussian distribution with an initial mean of 0 and a standard deviation of 0.05. The dropout rate of both embeddings was 0.2 or 0.5. The dimension of word embedding was 300, and that of Pinyin characters was 256. Empirically, A separable convolutional layer of 256 convolutions of size 3 was employed in the word and Pinyin character encoder. We used an Extractor in Pinyin encoder and 3 Extractors in combination part. All Extractors owned the same setup: There were 256 input channels, 4 heads in the attention block. Thus, separable convolutional layers of 64 convolutions of size 3 were applied to generate $\mathbf{Q}_{:,:,i}, \mathbf{K}_{:,:,i}, \mathbf{V}_{:,:,i}$, and a separable convolutional layer of 256 convolutions of size 3 were used at the end of the attention block. The max pooling with size 3 and stride 2 was used in the extraction block. It is similar to the max pooling in Pinyin character encoder. After max pooling, there were 2 separable convolutional layers with the setup as same as that in encoder parts. We used 1 or 3 full-connected layers with size 256 and dropout rate 0.2 or 0.5 in the classifier. Moreover, we employed Adam optimizer with an initial learning rate of 0.001. The loss function was the cross-entropy. All experiments were implemented using Keras and were performed on GPU 1080Ti.

5 RESULTS AND ANALYSIS

Testing Error Rates The testing error rate results are shown in Table 3. Due to the page limit, we list the best results of their variations with different hyperparameters. The results of EAN with the hybrid encoder or with the single word encoder are better than the state-of-the-art baseline methods (including TAN) on all datasets. It shows that EAN excels in the accuracy of Chinese text classification and the Extractor is very powerful to capture long-range dependencies or global features. And the EAN with the hybrid encoder performs better than EAN with the single encoder on all data sets, which proves the advantage of the hybrid encoder. We also observe that TAN with the hybrid

[1] https://github.com/zhangxiangxiao/glyph

Table 3: Testing error rates on benchmark datasets, in percentage. We ran EAN and TAN on 5 datasets. All other results are directly cited from the respective papers.

Method	Encoder	Dianping	Ifeng	JD.f	JD.b	Chinanews
EmbedNet [28]	Word	24.55	20.73	50.05	10.37	14.75
	Pinyin.character	25.42	19.21	48.75	9.46	11.84
	Pinyin. word	23.70	19.46	49.15	9.58	11.92
	Character	23.60	17.01	48.29	9.41	11.04
	Byte	24.09	17.12	48.56	9.19	10.55
GlyphNet [28]	-	24.31	18.02	48.97	9.85	12.26
OnehotNet [28]	Byte	23.17	16.49	48.10	9.31	10.62
	Pinyin	23.53	18.90	48.42	9.49	11.71
	Word	23.03	18.30	48.30	8.82	10.76
Linear model [28]	Pinyin. word	23.35	22.38	48.47	8.98	13.98
	Character	23.59	21.52	48.18	8.92	13.37
	Word	22.62	16.65	48.11	9.11	9.24
fastText [28]	Pinyin. word	22.42	17.86	48.13	8.73	9.39
	Character	22.34	16.31	47.99	8.72	9.10
	Word	22.62	16.65	48.11	9.11	9.24
TAN	Word	22.21	16.02	47.69	8.31	9.31
	Hybrid	22.08	14.95	46.82	8.31	9.00
EAN	Word	22.16	15.89	47.72	8.61	9.02
	Hybrid	**21.46**	**14.72**	**46.63**	**8.10**	**8.77**

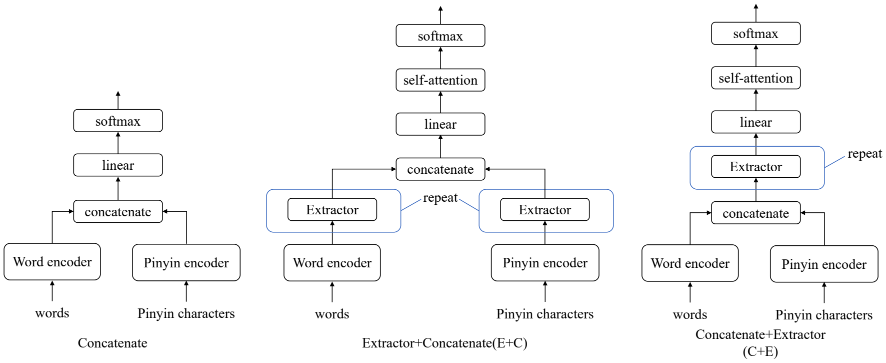

Figure 2: The different combination methods architecture.

encoder or the single word encoder are also better than the state-of-the-art baseline methods from Zhang & LeCun (2017) on all datasets. It proves that Transformer can obtain good results in Chinese text classification, as it has done in other NLP tasks.

Model Variations To evaluate the impact of different hyperparameters of EAN, we tested several EAN or TAN variations on JD.b and Ifeng. The variations contained different combination methods (only EAN), a different number of the Extractor or Transformer, and a different number of heads. We design 2 extra combination methods to prove the effectiveness of our combination method, which is shown in Figure 2. The concatenate combination method remove Extractors and self-attention after the concatenate layer (simple features concatenation), and the Extractor+Concatenate (E+C) combination method place Extractors after the word encoder and the Pinyin encoder respectively (the complicated encoders with straightforward features combination). The Concatenate+Extractor (C+E) combination method is our combination method. The results are shown in Table 4.

In Table 4 rows 1, 2, and 4, we observe that the C+E combination method is better than other combination methods, and the E+C combination method is better than the concatenate combination method. It means that the simple features concatenation without attention structures is relatively difficult to capture the associations across encoders, and the straightforward encoders with complicated features combination may work better in comparison to the complicated encoders with straightforward features combination.The results of TAN are very close to those of EAN, but most results of letter are better than those of former.The model which obtain the optimal results contained concatenate+Extractor method, 3 Extractors(Transformers) with 4 heads.

Table 4: Testing error rates for different model variations, in percentage. E+C indicates Extractor+Concatenate. C+E indicates Concatenate+Extractor. The Extractor in Pinyin encoder has not yet been calculated. All variations contained hybrid encoder.

Extractor(Transformer)	Head	Combination method	Ifeng (EAN)	JD.b (EAN)	Ifeng (TAN)	JD.b (TAN)
3	4	Concatenate	19.78	8.57	-	-
3+3	4	E+C	15.43	8.38	-	-
2	4	C+E	14.91	8.31	15.14	8.42
3	4	C+E	**14.72**	**8.10**	14.95	8.23
4	4	C+E	15.05	8.28	14.98	8.31
5	4	C+E	15.00	8.19	15.06	8.26
3	2	C+E	15.57	8.28	15.61	8.30
3	6	C+E	15.16	8.33	15.05	8.29
3	8	C+E	15.46	8.42	15.48	8.48

Table 5: Comparison of model parameters. E+C indicates Extractor+Concatenate. C+E indicates Concatenate+Extractor.

Methods		Parameters
VDCNN	9-layers	2.2 M
	17-layers	4.3 M
	29-layers	4.6 M
	49-layers	7.8 M
Bi-BloSAN	-	3.6M
TAN	Word	2.51 M
	Hybrid	3.39 M
EAN	C+E (Word)	1.37 M
	C+E (Hybrid)	1.87 M
	Concatenate (Hybrid)	9.61 M-16.95 M
	E+C (Hybrid)	2.68 M-2.69 M

Parameters We compared the parameters with some text classification model including Bi-BloSAN, VDCNN, TAN, and EAN with different combination methods. The results are shown in Table 5. There are diiferent parameters of EAN due to diiferent word and character lengths on different datasets, especially Concatenate and E+C combation methods. The parameters of EAN (C+E) are fewer than other models, which shows the excellent property of the Extractor. specifically, the parameters of EAN (C+E) are fewer than the parameters of TAN because we use the separable CNN to replace the linear operation such as full-connection layers and employ downsampling operation like the max pooling layer to compress feature maps. In fact, feature maps are halved in each extraction block of the Extractor. Moreover, the parameters of EAN (C+E) are much fewer than EAN (Concatenate) and EAN (E+C). It means our combination method is computationally relatively cheaper. Thus, as we mentioned before, the feature maps of EAN are narrow but enough to obtain a good result. It can alleviate the efficiency problem of the hybrid encoder method such as too many parameters or too slow speed.

6 Conclusions

This paper proposes a novel attention network, the Extractor-attention network (EAN), for Chinese text classification. Compared to the traditional Chinese text classification methods using only word encoder, our approach uses hybrid encoder including words and Pinyin characters, which takes full advantage of the extra Pinyin information to improve the performance. Moreover, there is a new structure named the Extractor in our work, reduces the number of parameters in EAN and makes it excellent to extract feature. Thus, EAN obtains the state of the art results on 5 public Chinese text classification datasets. Finally, we also analyze the effects of different encoders structures on the method.

References

Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seungwon Hwang. Cold-start aware user and product attention for sentiment classification. In *56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pp. 2535–2544. Association for Computational Linguistics (ACL), 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *stat*, 1050:21, 2016.

Phil Blunsom, Edward Grefenstette, and Nal Kalchbrenner. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational . . . , 2014.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Alexis Conneau, Holger Schwenk, Yann LeCun, and Löc Barrault. Very deep convolutional networks for text classification. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017*, pp. 1107–1116. Association for Computational Linguistics (ACL), 2017.

Yue Gu, Kangning Yang, Shiyu Fu, Shuhong Chen, Xinyu Li, and Ivan Marsic. Multimodal affective analysis using hierarchical attention strategy with word-level alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2225–2235, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 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*, pp. 427–431, 2017.

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

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

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pp. 1188–1196, 2014.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 138–143, 2018.

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

Pengfei Liu, Ji Zhang, Cane Wing-Ki Leung, Chao He, and Thomas L Griffiths. Exploiting effective representations for chinese sentiment analysis using a multi-channel convolutional neural network. *arXiv preprint arXiv:1808.02961*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86. Association for Computational Linguistics, 2002.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. Radical embedding: Delving deeper to chinese radicals. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 594–598, 2015.

Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2321–2331, 2018.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489, 2016.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 286–291, 2017.

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

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

Hang Zhuang, Chao Wang, Changlong Li, Qingfeng Wang, and Xuehai Zhou. Natural language processing service based on stroke-level convolutional networks for chinese text classification. In *2017 IEEE International Conference on Web Services (ICWS)*, pp. 404–411. IEEE, 2017.

Hang Zhuang, Chao Wang, Changlong Li, Yijing Li, Qingfeng Wang, and Xuehai Zhou. Chinese language processing based on stroke representation and multidimensional representation. *IEEE Access*, 6:41928–41941, 2018.