A Radical-aware Attention-based Model for Chinese Text Classification

Hanqing Tao[†], Shiwei Tong[†], Hongke Zhao[†], Tong Xu[†§*], Binbin Jin[†], Qi Liu[†§]

[†]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China
[§]School of Data Science, University of Science and Technology of China
{hqtao, tongsw, zhhk, bb0725}@mail.ustc.edu.cn, {tongxu, qiliuql}@ustc.edu.cn

Abstract

Recent years, Chinese text classification has attracted more and more research attention. However, most existing techniques which specifically aim at English materials may lose effectiveness on this task due to the huge difference between Chinese and English. Actually, as a special kind of hieroglyphics, Chinese characters and radicals are semantically useful but still unexplored in the task of text classification. To that end, in this paper, we first analyze the motives of using multiple granularity features to represent a Chinese text by inspecting the characteristics of radicals, characters and words. For better representing the Chinese text and then implementing Chinese text classification, we propose a novel *Radical-aware Attention-based Four-Granularity* (RAFG) model to take full advantages of Chinese characters, words, character-level radicals, word-level radicals simultaneously. Specifically, RAFG applies a serialized BLSTM structure which is context-aware and able to capture the long-range information to model the *character sharing* property of Chinese and sequence characteristics in texts. Further, we design an attention mechanism to enhance the effects of radicals thus model the *radical sharing* property when integrating granularities. Finally, we conduct extensive experiments, where the experimental results not only show the superiority of our model, but also validate the effectiveness of radicals in the task of Chinese text classification.

Introduction

Text classification which aims to select the most appropriate assignment to an untagged text from a predefined set of tags has been widely studied (Peng et al. 2003). However, most existing studies on text classification are professionally conducted for English. Recently, the classification of Chinese text has attracted more and more attention.

Chinese, a language derived from pictographs, is essentially different from English or other phonetic languages. For Chinese, one of the most unique is that the character system of Chinese is based on hieroglyphics, which has the raw meanings. That is to say, not only words and characters can express specific meanings, but also radicals are important carriers of semantics. This special property is an important difference between Chinese and English text classification.

---

[*]Corresponding author

Glyph Origin	Radical (Chinese Characters)	English
亻	亻 (仆, 伴)	Man (servant, partner)
⌀	目 (看, 瞳)	Eye (look, pupil)
才	扌 (打, 挖)	Hand (hit, dig)
⊖	日 (晴, 暗)	Sun (sunny, dark)
�	雨 (雾, 霜)	Rain (fog, frost)
ᨑ	山 (峰, 崖)	Mountain (peak, cliff)

Figure 1: Glyph origin of some radicals, and the semantic connection between Chinese characters and radicals.

As shown in Figure 1, radical is a kind of semantic unit with some graphic characteristics. A radical is often related to some certain concepts, e.g., we use "Eye" to "look"; we use "Hand" to "hit" or "dig"; "sunny" and "dark" are relevant to the light of "Sun"; the "peak" and "cliff" are places of a "Mountain". From these examples, we can preliminarily see that radicals might help us to recognize semantics. Although some scholars have realized the importance of radicals in representing individual Chinese words, few of existing studies have exploited radicals to help classify Chinese texts.

Actually, there are two special kinds of properties in Chinese, which have not been systematically explored. The first property is *radical sharing*. As illustrated above, the meaning of a Chinese character can be partly expressed through its radical. If several Chinese characters share a common radical, that radical is usually the core semantic association between them. As shown in Table 1, those five Chinese characters share a common radical "insect", and in fact they are indeed corresponding to five different kinds of insects, which reflects the important role of radicals in terms of character semantics. Correspondingly, the other property is *character sharing*. That is, the meaning of a Chinese word can be expressed through its containing characters. If several Chinese words have one character in common, that character is also usually the core semantic association between these words (see Table 2). Inspired by the importance of radicals in Chinese and these two properties, in this study, we propose to exploit the utilities of *character-level radicals* and *charac-*

Table 1: Characters with the same radical "insect".

Chinese Characters	Radical	English
蝇	虫	fly
蚊	虫	mosquito
蜂	虫	bee
虱	虫	louse
蚁	虫	ant

Table 2: Words with the same character "cattle".

Chinese Words	Chinese Characters	English
公牛	公 (male) + 牛 (cattle)	bull
母牛	母 (female) + 牛 (cattle)	cow
牛奶	牛 (cattle) + 奶 (milk)	milk
牛肉	牛 (cattle) + 肉 (meat)	beef
牛角	牛 (cattle) + 角 (horn)	horn

ters for the task of Chinese text classification.

Additionally, in most current text classification methods especially those aimed at English materials, words are regarded as the main features. Because there are natural spaces between English words as delimiters, it seems quite normal and intuitive. Unfortunately, there are no such delimiters in Chinese, thus the definition of "word" granularity in Chinese is unclear, which is another difference between Chinese and English text classification. In order to make methods originally developed for English adapted to work with Chinese, segmentation into the form of word unit is desirable (Liu et al. 2007). Therefore, there has been a lot of researches on Chinese Word Segmentation (CWS) aiming to segment Chinese texts into word sequences (Peng, Feng, and McCallum 2004). Following these advances and considering the utilities of radicals, we also exploit *words* and *word-level radicals* for Chinese text classification in this study.

Specifically, in this paper, we present an explorative study on Chinese text classification with a special focus on the utilization of radicals. First, to explore the characteristics of radicals, we propose to subdivide radicals into character-level radicals and word-level radicals. Then, in order to better represent the Chinese text, we propose a *Radical-aware Attention-based Four-Granularity* (RAFG) model to jointly leverage four granularities of features, i.e., Chinese *characters*, *words*, *character-level radicals* and *word-level radicals* to implement Chinese text classification. Additionally, to model the *character sharing* property of Chinese and sequence characteristics in texts, RAFG applies a serialized BLSTM structure which is context-aware and able to capture the long-range information. Further, we design an attention mechanism to enhance the effects of radicals thus model the *radical sharing* property when integrating granularities. Finally, we conduct extensive experiments on two real-world datasets. The experimental results not only show the superiority of our methods, but also demonstrate the effectiveness of radicals in the task of Chinese text classification.

Related Work

Text classification is an important task of text mining (Qin et al. 2018), where machine learning algorithms such as Logistic Regression, Decision Trees, Naïve Bayes and SVM are widely used (Hotho, Nürnberger, and Paaß 2005). Specifically, text representation is a key problem in text classification. Initially, the Bag-Of-Words (BOW) representation model was one of the most common text representation models. With the rise of Deep Neural Networks, word embedding is proposed to tackle the problem of *curse of di-*

mensionality (Bengio et al. 2003). Further, pre-trained word embeddings were proved to be effective in representing sentences (Mikolov, Yih, and Zweig 2013). Afterwards, to deal with the shortcoming of ignoring the order of words, sequence representation models such as Recurrent Neural Network (Hochreiter and Schmidhuber 1997; Graves, Mohamed, and Hinton 2013; Chung et al. 2014) and Convolutional Neural Network (CNN) (Kim 2014; Liu et al. 2018) were then proposed. After that, some novel models such as structure-enhanced LSTMs (Zhu, Sobihani, and Guo 2015; Tai, Socher, and Manning 2015) and the combination of previous models (Lai et al. 2015) were proposed one after another. Indeed, attention mechanism is a novel and effective technique which has been widely used in Natural Language Processing recently (Vaswani et al. 2017). It shows its superiority in many fields such as document classification (Yang et al. 2016), sentiment classification (Zhou, Wan, and Xiao 2016), sentence representation (Lin et al. 2017; Huang et al. 2017) and so on.

In this area, some scholars have realized the specificity of Chinese. However, most existing researches about modeling the characteristics of Chinese and radicals mainly focus on the embedding problem for words or characters. In view of the uniqueness of radicals, Sun et al. (2014) first proposed to utilize radical information to improve Chinese character embedding. After that, Chen et al. (2015) argued that the semantic meaning of a word was also related to the meanings of its composing characters, and the word embeddings could be enhanced with the help of the context characters. After that, Shi et al. (2015) made a tentative exploration about radicals, and demonstrated the utility of radicals in some conditions. Furthermore, some methods were proposed to use radical information to strengthen Chinese word embedding (Yin et al. 2016; Yu et al. 2017), but the scope of their research on radicals was still limited to the embedding problem. Afterwards, Peng et al. (2017) proposed to utilize radicals to make sentiment analysis for sentences, which provided some new insights for the utility of radicals.

Different from previous work, our goal is to take advantages of radicals and leverage four different granularities of features to comprehensively model Chinese texts. Furthermore, we systematically integrate these features into the task of Chinese text classification, so that to deal with the huge difference between Chinese and English.

Methodology

In this section, we first formally introduce the Chinese text classification problem, then we describe the technical detail-
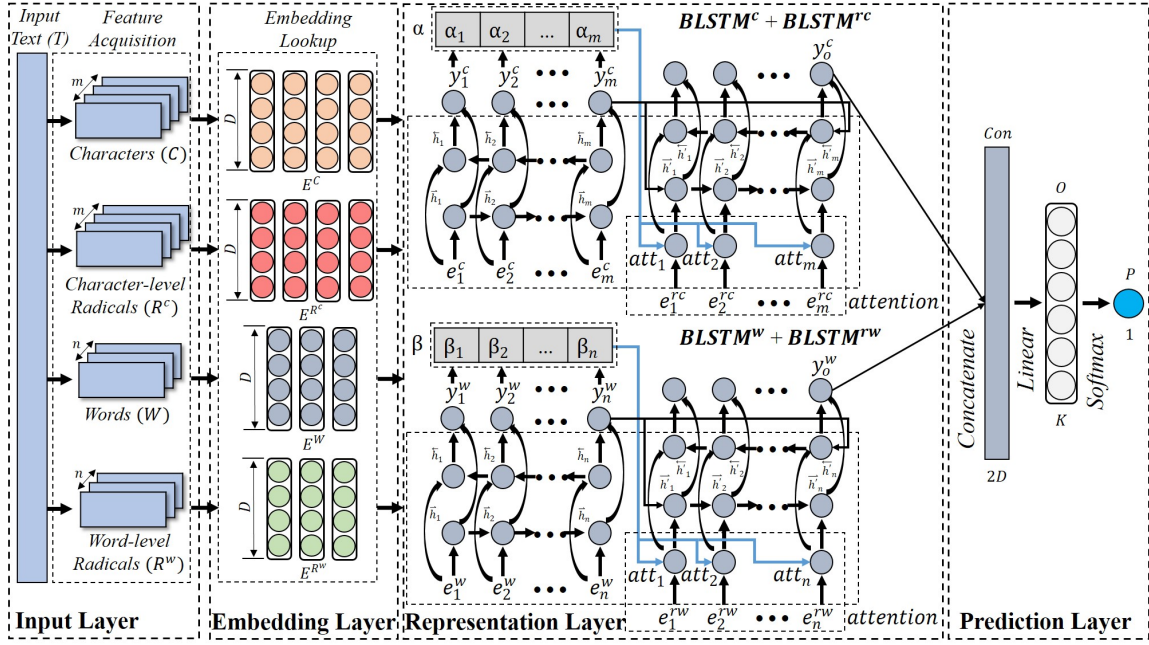
Figure 2: *Radical-aware Attention-based Four-Granularity* (RAFG) model.

s of our *Radical-Aware Attention-based Four-Granularity* model (RAFG). Finally, we discuss the training of RAFG.

Problem Overview

Generally, the task of text classification is to select the most appropriate assignment P to an untagged text T from a pre-defined set of tags or labels U, i.e., for a text classification system, its input is T, and the output is a prediction $P \in U$. More formally, the task is to learn a classification function:

$$\mathcal{F}(T) \to P.$$

Technical Details of RAFG Model

As shown in Figure 2, the aim of our work is to utilize four different granularities of features to comprehensively model Chinese texts, so as to further realize the classification task of Chinese texts. Overall, RAFG contains four parts: *Input Layer*, *Embedding Layer*, *Representation Layer* and *Prediction Layer*. The details are as follows.

Input Layer mainly tackles the problem of *Feature Acquisition* of input text. Figure 3 gives a graphical illustration about RAFG to get the four-granularity features of a Chinese text. For a Chinese raw text T, it contains m characters, i.e., $C = \{c_1, c_2, ..., c_m\}$, where each character c_i ($1 \leqslant i \leqslant m$) is an independent item. Meanwhile, T will be cut into n words $W = \{w_1, w_2, ..., w_n\}$. Since a word can often be divided into several characters, it is obvious that $n \leqslant m$. Then, the characters and words will be mapped into two kinds of radicals respectively by looking up *Xinhua dictionary*, i.e., m character-level radicals $R^c = \{r_1^c, r_2^c, ..., r_m^c\}$ and n word-level radicals $R^w = \{r_1^w, r_2^w, ..., r_n^w\}$. After the processing, the four-granularity features of T are obtained.
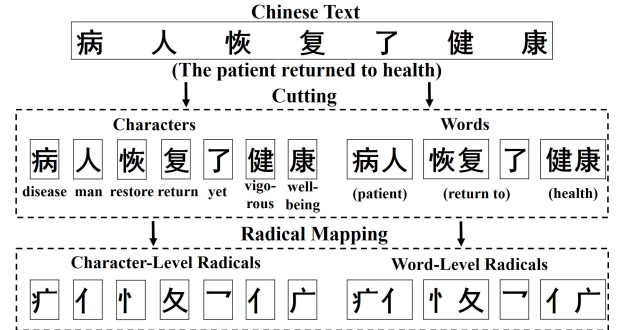

Figure 3: Getting the four-granularity features of a Chinese text.

In the mapping process of character-level radicals, each digit, punctuation, or each letter in a word will be mapped to a "-" respectively (e.g., "sun" consists of three letters "s", "u" and "n", so that "sun" will be mapped into three "-"). In the mapping process of word-level radicals, each non-Chinese item will be mapped to a single "-" to indicate that it does not have any radicals (e.g., "2019", "!" and "sun" will be mapped to a single "-" respectively). Thus, the length of C is equal to R^c, and the length of W is equal to R^w, i.e., $|C| = |R^c|$, $|W| = |R^w|$.

Embedding Layer aims to represent each item from Input Layer in a continuous space. It receives four granularities of features (i.e., C, W, R^c, R^w) and outputs four embedding matrices by looking up embedding dictionary. As mentioned before, the lengths of the four-granularity features satisfy $|C| = |R^c|$ and $|W| = |R^w|$. To simplify the prob-

lem, we set the vector dimension of each Chinese character, word, character-level radical and word-level radical to the same size D. Thus, a Chinese text can be represented by four vector sequences, i.e., $E^C = \{e_1^c, e_2^c, ..., e_m^c\}$, $E^W = \{e_1^w, e_2^w, ..., e_n^w\}$, $E^{R^c} = \{e_1^{rc}, e_2^{rc}, ..., e_m^{rc}\}$ and $E^{R^w} = \{e_1^{rw}, e_2^{rw}, ..., e_n^{rw}\}$. Exactly, these four vector sequences are also four embedding matrices, i.e., $E^C \in R^{m \times D}$, $E^W \in R^{n \times D}$, $E^{R^c} \in R^{m \times D}$ and $E^{R^w} \in R^{n \times D}$.

Representation Layer aims to generate a comprehensive representation of input text T by combining the context and radicals information together. Corresponding to the property of *character sharing*, the recurrent structure of LSTM naturally processes words and characters one by one, which not only memorizes the characters or words that have already appeared, but also deals with the problem of vague definition of Chinese words in segmentation to some extent (Peng, Feng, and McCallum 2004). In view of this advantage, we utilize an implementation of LSTM proposed by (Graves, Mohamed, and Hinton 2013) and apply the bidirectional setting (i.e., BLSTM) to capture both the forward and backward context information. Formally, given a specific feature embedding sequence of a sentence $s = \{x_1, x_2, ..., x_N\}$, the hidden vector of a BLSTM is calculated as follows:

$$
\begin{aligned}
\overrightarrow{h_t} &= LSTM(\overrightarrow{h}_{t-1}, x_t), \\
\overleftarrow{h_t} &= LSTM(\overleftarrow{h}_{t+1}, x_t), \\
y_t &= [\overrightarrow{h_t}, \overleftarrow{h_t}],
\end{aligned}
\tag{1}
$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ is the forward hidden vector and backward hidden vector respectively at the t-th step in the BLSTM. And y_t is the hidden output of each BLSTM at the t-th step, which is the concatenation of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$.

As shown in Figure 2, there are two serialized BLSTMs in the representation layer (i.e., $BLSTM^c + BLSTM^{rc}$ and $BLSTM^w + BLSTM^{rw}$). In $BLSTM^c$ and $BLSTM^w$, the values of their initial hidden states are set to zero. Meanwhile, $BLSTM^{rc}$ and $BLSTM^{rw}$ receive the last hidden states of $BLSTM^c$ and $BLSTM^w$ as input respectively, which allows the context information of characters and words can be further combined with the context information of character-level radicals and word-level radicals.

Additionally, to assign important weights to certain radicals thus model the *radical sharing* property when integrating granularities, we design an attention mechanism which can capture the interrelations between radicals and their corresponding characters or words. Everytime $BLSTM^{rc}$ or $BLSTM^{rw}$ receives a vector embedding of a radical (i.e., e_i^{rc} or e_j^{rw}), each $y_\epsilon^c \in Y^c = \{y_1^c, y_2^c, ..., y_m^c\}$ and $y_\theta^w \in Y^w = \{y_1^w, y_2^w, ..., y_n^w\}$ will conduct the dot product operation with e_i^{rc} and e_j^{rw} respectively. Thus, the attention vector α' for e_i^{rc}, β' for e_j^{rw} is obtained as follows:

$$
\begin{aligned}
\alpha' &= [\alpha_1', ..., \alpha_\epsilon', ..., \alpha_m'], \ \alpha_\epsilon' = f(y_\epsilon^c, e_i^{rc}), 1 \leqslant \epsilon \leqslant m, \\
\beta' &= [\beta_1', ..., \beta_\theta', ..., \beta_n'], \ \beta_\theta' = f(y_\theta^w, e_j^{rw}), 1 \leqslant \theta \leqslant n,
\end{aligned}
\tag{2}
$$

where α_ϵ' and β_θ' denote the ϵ-th weight of a character-level radical or the θ-th weight of a word-level radical respec-

tively, and $f(a, b)$ denotes the dot product function. But before the weighted sum operation, we need to normalize these weights using the softmax function, i.e., α_i and β_j are obtained as follows:

$$
\begin{aligned}
\alpha_i &= \frac{exp(\alpha_\epsilon')}{\sum_{\epsilon=1}^m exp(\alpha_\epsilon')}, \ where \sum_{i=1}^m \alpha_i = 1, \\
\beta_j &= \frac{exp(\beta_\theta')}{\sum_{\theta=1}^n exp(\beta_\theta')}, \ where \sum_{j=1}^n \beta_j = 1,
\end{aligned}
\tag{3}
$$

then the vector embedding of r_i^c and r_j^w will be modified as:

$$
\widetilde{e}_i^{rc} = \sum_{\epsilon=1}^m \alpha_\epsilon y_\epsilon^c, \ \widetilde{e}_j^{rw} = \sum_{\theta=1}^n \beta_\theta y_\theta^w,
\tag{4}
$$

where y_ϵ^c denotes the ϵ-th item of Y^c, and y_θ^w denotes the θ-th item of Y^w. After the attention operation (i.e., att_i in Figure 2), $\widetilde{e}_i^{rc}$ and $\widetilde{e}_j^{rw}$ have fused the weight information of character context and word context respectively. Then, $BLSTM^{rc}$ and $BLSTM^{rw}$ will further learn the contextual information of $\widetilde{e}_i^{rc}$ and $\widetilde{e}_j^{rw}$ through the calculations described in Equation (1).

Prediction Layer. As a result, we take the final hidden layer states of $BLSTM^{rc}$ and $BLSTM^{rw}$ (i.e., y_o^c and y_o^w) as the final output, then we concatenate them together into a comprehensive representation $Con \in R^{2D}$. Here, Con is exactly the ultimate representation of input text T. After that, we feed Con into a fully-connected neural network to get an output vector $O \in R^K$ (K is the number of classes, i.e., $K = |U|$):

$$
O = sigmoid(Con \times W),
\tag{5}
$$

where $W \in R^{2D \times K}$ is the weight matrix for dimension transformation, and $sigmoid(\cdot)$ is a non-linear activation function. Finally, we apply a softmax layer to map each value in O to conditional probability and realize the classification as follows:

$$
P = argmax(softmax(O)).
\tag{6}
$$

Model Training. Since what we are trying to solve is a multi-class classification task, we follow the work in (Zhou et al. 2016) to apply the cross-entropy loss function to train our model, and the goal is to minimize the following *Loss*:

$$
Loss = -\sum_{T \in Corpus} \sum_{i=1}^K p_i(T) \log p_i(T),
\tag{7}
$$

where T is the input text, *Corpus* denotes the training corpus and K is the number of classes. In the training process, we apply *Adagrad* as optimizer to update the parameters of RAFG, including W and all parameters (weights and biases) in each BLSTM. To avoid the overfitting problem, we apply the dropout mechanism at the end of the embedding layer.

Experiments

Dataset Preparation

Dataset#1. To fit the problems studied in this paper, we choose a public Chinese text dataset (Zhou et al. 2016)

which is suitable for our work. It contains 47,952 Chinese news titles with 32 gold standard classification labels for training and 15,986 titles for testing. In order to preserve the raw information of the dataset and validate the robustness of our methods, we do not intend to filter out any texts.

Dataset#2. To test the characteristics and importance of radicals, we filter the original *dataset#1* by removing all texts whose non-Chinese ratio is larger than 20% since each non-Chinese item does not have a radical (e.g., for a text whose length is n_0 and the number of non-Chinese items is n_1, the non-Chinese ratio is computed as n_1/n_0). Those texts which contain special characters are also removed (e.g. "\u3000"). After the processing, we still have more than 75% of the raw data: 36,431 texts for training and 12,267 texts for testing. Table 3 shows the statistics of *dataset#1* and *dataset#2*.

Experimental Setup

Xinhua Dictionary Setting. Since we need to map each Chinese character to a radical, we use a Xinhua dictionary dataset[1] to achieve this goal. The Xinhua dictionary dataset covers all Chinese characters and radicals appeared in the datasets. There are 20,849 Chinese characters and 270 kinds of radicals in it. Due to the need of the radical mapping step, we have artificially added a "-" to map non-Chinese items that do not have a radical. In conclusion, the total number of radicals is 271.

Embedding Setting. In this study, we use *jieba*[2] as the word segmentation tool to cut Chinese texts into word sequences. Considering the performance of deep learning model is highly related with the quality of embedding vectors, we apply a well pre-trained word embedding model based on a large corpora[3] to represent the words, which is comprehensive in contents. But for Chinese characters and radicals, there are no ready-made models available. To tackle this problem, we apply the public word2vec tool (*Gensim*[4]) to train embeddings for characters, character-level radicals and word-level radicals. The dimension of those embeddings are all set to 256 (i.e., $D = 256$). As mentioned before, each non-Chinese item will be mapped to a "-" in the radical mapping process, so "-" will be randomized with a 256 dimensional vector and tuned during the training process. It should be pointed out that *dataset#1* and *dataset#2* will produce corresponding two sets of embeddings for characters, character-level radicals and word-level radicals. In addition, we implement our neural network models using MXNet[5], with several GPUs accelerating the experimental process.

Training Setting. In RAFG, we empirically set the dimension of hidden vectors of each BLSTM to 256. To avoid overfitting, when we get the embeddings of characters, words, character-level radicals and word-level radicals, we drop 50% of them. In addition, we have tried some learning rates and finally set the learning rate to 0.03, which can guarantee the speed of training on the one hand, and prevent

---

[1] https://pan.baidu.com/s/1TJcrFFxF0xLHuHKRC9XHMA

[2] https://github.com/fxsjy/jieba

[3] https://spaces.ac.cn/archives/4304

[4] http://radimrehurek.com/gensim/

[5] https://mxnet.apache.org/

Table 3: Statistics of *dataset#1* and *dataset#2*.

Dataset		Count	Len (Avg. / Max)	Class
Dataset#1	Train	47,952	17.8 / 56	32
	Test	15,986	17.7 / 56	
Dataset#2	Train	36,431	16.7 / 46	32
	Test	12,267	16.7 / 43	

vibration on the other hand. Furthermore, we set the batch-size to 32 and the epoch of training process to 200. Finally, we use $Precision$ (P), $Recall$ (R) and $F_1\text{-}measure$ (F_1) to evaluate the performance (Hotho, Nürnberger, and Paaß 2005; Qiao et al. 2019):

$$F_1 = \frac{2PR}{P + R}. \qquad (8)$$

Baseline Methods

Since there are few works at present to systematically analyze the characteristics of Chinese and radicals, we have designed several sets of comparative experiments to verify the effectiveness of radicals for Chinese text classification. It is the results that guide the design of RAFG and the conduct of subsequent experiments.

- **SVM + BOW**. To verify whether the radicals are useful, we use tf-idf weights of Chinese characters (C), words (W), character-level radicals (R^c), word-level radicals (R^w) as features respectively, and apply the Bag-Of-Words (BOW) method to train *liblinear SVM* classifier[6].

- **LSTM / Four LSTMs** (Hochreiter and Schmidhuber 1997; Graves, Mohamed, and Hinton 2013). We use a single LSTM to process Chinese words and characters respectively as two baselines, i.e., LSTM (E^W) and LSTM (E^C). To preliminarily test the performance of integrating radical features, we use Four LSTMs as a whole to respectively process E^W, E^C, E^{R^w} and E^{R^c} as an explorative baseline, where four corresponding hidden output vectors will be concatenated into a vector like Con in RAFG.

- **Four BLSTMs**. Corresponding to Four LSTMs, we use Four BLSTMs as another baseline to verify the effectiveness of bidirectional setting.

- **C-LSTMs / C-BLSTMs** (Zhou et al. 2016). C-LSTMs applies two independent LSTMs to concatenate word and character features. And C-BLSTMs is the bidirectional version of C-LSTMs. Compared with C-BLSTMs, RAFG is exactly a further improvement which properly takes the information of extra two kinds of radicals into account.

Experimental Results

The experimental results are shown in Table 4. We can notice that our model (RAFG) gains a higher performance than any other comparison methods. To figure out the internal causes, we carry out the following detailed analysis.

- By comparing the experimental results of SVM + BOW using Chinese words, characters, word-level radicals and

---

[6] https://www.csie.ntu.edu.tw/~cjlin/liblinear/

Table 4: Experimental results of different methods on $dataset\#1$ and $dataset\#2$.

Methods	Dataset#1 F_1 (P, R)	Dataset#2 F_1 (P, R)
SVM + BOW (W)	0.7552 (0.7639, 0.7514)	0.7341 (0.7459, 0.7303)
SVM + BOW (C)	0.7421 (0.7440, 0.7420)	0.7252 (0.7268, 0.7255)
SVM + BOW (R^w)	0.6834 (0.6913, 0.6800)	0.6762 (0.6858, 0.6729)
SVM + BOW (R^c)	0.4697 (0.4652, 0.4809)	0.4691 (0.4636, 0.4813)
LSTM (E^C)	0.7077 (0.7108, 0.7077)	0.6871 (0.6926, 0.6887)
LSTM (E^W)	0.8029 (0.8034, 0.8031)	0.7875 (0.7893, 0.7885)
Four LSTMs ($E^W + E^C + E^{R^w} + E^{R^c}$)	0.8072 (0.8078, 0.8074)	0.7904 (0.7912, 0.7910)
Four BLSTMs ($E^W + E^C + E^{R^w} + E^{R^c}$)	0.8098 (0.8103, 0.8103)	0.7915 (0.7925, 0.7921)
C-LSTMs ($E^W + E^C$)	0.8112 (0.8118, 0.8115)	0.7931 (0.7944, 0.7929)
C-BLSTMs ($E^W + E^C$)	0.8128 (0.8135, 0.8131)	0.7956 (0.7951, 0.7972)
Ours (RAFG)	**0.8181** (0.8181, 0.8187)	**0.7999** (0.7993, 0.8010)

character-level radicals as features respectively, we can find that SVM + BOW (W) achieves the best performance, and SVM + BOW (C) is second to it. In other words, Chinese words and characters are unquestionably important semantic features in terms of Chinese text classification. At the same time, the performance of SVM + BOW (R^w) is worth paying attention to, which reveals the fact that word-level radicals in Chinese are semantically useful. In addition, the performance of SVM + BOW (R^c) is much lower than that of word-level radicals, which proves that a radical alone is not able to reflect enough semantics, and it is only when they are combined or placed in a certain order that useful meanings can be expressed.

- When comparing LSTM (E^C) and LSTM (E^W) with four SVM + BOW baselines, we can see that LSTM (E^W) achieves better performance on both *dataset#1* and *dataset#2*, which takes the contextual information and the order of words into account. At the same time, the results reveal that Chinese words are more important than Chinese characters to some extent. In addition, the results of C-LSTMs ($E^W + E^C$) is better than that of LSTMs (E^W), which not only indicates that it is better to utilize words and characters together, but also shows that words and characters are mutually reinforcing.

- For LSTM (E^C), LSTM (E^W), Four LSTMs ($E^W + E^C + E^{R^w} + E^{R^c}$), Four BLSTMs ($E^W + E^C + E^{R^w} + E^{R^c}$), C-LSTMs ($E^W + E^C$) and C-BLSTMs ($E^W + E^C$), results on these two datasets show that the performance of bidirectional LSTM is a little better than that of single-direction LSTM. In addition, we can see that blindly applying LSTM (i.e., Four LSTMs and Four BLSTMs) to process radicals is not so effective, with a lower performance than that of C-LSTMs and C-BLSTMs. Particularly, some simple radicals nowadays cannot convey stable meanings due to the simplification of Chinese, and this maybe a reason for that simply introducing radicals to Chinese text classification cannot improve the performance well (i.e., Four LSTMs and Four BLSTMs). Some other special methods need to be adopted to make rational use of radicals. Out of this, the attention mechanism in RAFG allows the model to pay more attention to relatively important items in a certain text. The better per-

formance of RAFG exactly proves the rationality of our ideas and the effectiveness of attention mechanism.

Discussion

Here, in order to provide some intuitionistic and explanatory clues for the possible causes why our RAFG model outperforms the existing works, we tentatively analyze the tf-idf distributions of some randomly selected radicals under 32 kinds of texts. Obviously, the seven radicals in Figure 4 are corresponding to seven peaks of seven different classes, which supports our assumption that radicals can help recognize semantics and classify Chinese texts. For example, the original meaning of radical "clothing" is closed to the concept of class "dress", where the high tf-idf value is a convincing indication.
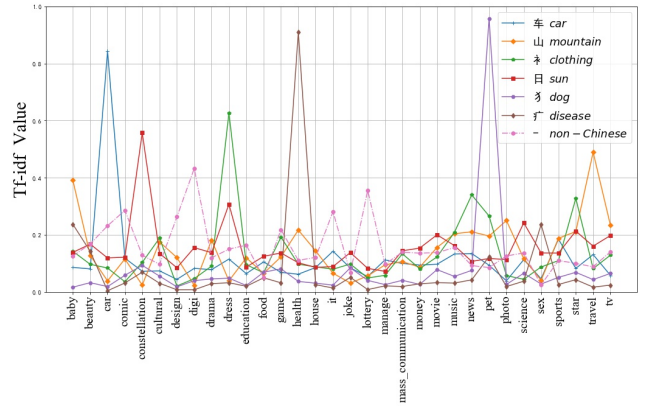

Figure 4: Tf-idf Distributions of Some Radicals.

In order to analyze the value of radicals in different conditions, we further compare RAFG with the most competitive baseline (i.e., C-BLSTMs) in all 32 classes. From Table 5, we can see that RAFG shows its superiority in most classes. However, in some specific classes (e.g., "food", "health", "photo", "sex"), C-BLSTMs gains a higher performance, which indicates that the role of radicals in the classification of these texts may not be so critical than that of words or characters. At the same time, we can find that in some

Table 5: Detailed comparison on *dataset#1* (left) and *dataset#2* (right).

Class (Num=32)	Train / Test	F_1-measure C-BLSMs	RAFG	Class (Num=32)	Train / Test	F_1-measure C-BLSMs	RAFG
baby	1466 / 534	0.9004	**0.9080**	baby	1314 / 482	0.9039	**0.9042**
beauty	1460 / 478	0.8768	**0.8857**	beauty	1057 / 363	0.8690	**0.8719**
car	1508 / 492	0.9037	**0.9058**	car	846 / 298	0.8727	**0.8769**
comic	1493 / 507	0.8186	**0.8276**	comic	758 / 246	0.7596	**0.7815**
constellation	1510 / 490	**0.9554**	0.9533	constellation	1329 / 425	0.9557	**0.9570**
cultural	1517 / 483	0.6393	**0.6499**	cultural	1360 / 441	0.6484	**0.6655**
design	1514 / 486	**0.8185**	0.8108	design	850 / 283	0.7717	**0.7748**
digi	1523 / 477	0.8566	**0.8740**	digi	227 / 66	0.6161	**0.6526**
drama	1504 / 496	0.7865	**0.7893**	drama	1338 / 434	0.7784	**0.7881**
dress	1471 / 529	0.8854	**0.8901**	dress	1106 / 397	0.8769	**0.8811**
education	1482 / 518	0.8862	**0.8963**	education	984 / 354	0.8591	**0.8663**
food	1504 / 496	**0.9565**	0.9520	food	1470 / 488	**0.9569**	0.9445
game	1492 / 508	0.8189	**0.8235**	game	923 / 322	0.8098	**0.8130**
health	1507 / 493	**0.9364**	0.9333	health	1244 / 405	**0.9360**	0.9266
house	1470 / 530	0.8427	**0.8484**	house	1219 / 462	**0.8520**	0.8511
it	1518 / 482	0.6520	**0.6709**	it	782 / 253	0.5578	**0.5838**
joke	1520 / 480	0.8647	**0.8667**	joke	1457 / 463	**0.8883**	0.8721
lottery	1530 / 470	**0.9801**	0.9776	lottery	214 / 54	0.8475	**0.8510**
manage	1499 / 501	0.7817	**0.7872**	manage	1311 / 443	0.7892	**0.7955**
mass_communication	1499 / 501	0.5712	**0.5835**	mass_communication	1182 / 412	0.5804	**0.5899**
money	1445 / 555	0.7778	**0.7910**	money	1140 / 428	0.7621	**0.7776**
movie	1510 / 490	0.7144	**0.7339**	movie	1227 / 415	**0.7274**	0.7100
music	1510 / 490	0.6759	**0.6864**	music	1159 / 380	0.6607	**0.6812**
news	1512 / 488	0.6703	**0.6876**	news	1379 / 447	0.6804	**0.6869**
pet	1500 / 500	0.8437	**0.8529**	pet	1372 / 450	0.8498	**0.8518**
photo	1512 / 488	**0.8883**	0.8818	photo	1292 / 403	**0.9085**	0.9007
science	1493 / 507	0.8339	**0.8346**	science	1210 / 391	0.8285	**0.8339**
sex	1505 / 495	**0.9386**	0.9344	sex	1488 / 493	**0.9405**	0.9376
sports	1491 / 509	0.8693	**0.8794**	sports	1258 / 433	0.8672	**0.8814**
star	1512 / 488	0.6543	**0.6622**	star	1356 / 429	**0.6706**	0.6611
travel	1501 / 499	**0.7770**	0.7759	travel	1355 / 444	0.7937	**0.7961**
tv	1474 / 526	**0.6343**	0.6234	tv	1224 / 463	**0.6418**	0.6301
Average		0.8128	**0.8181**	Average		0.7956	**0.7999**

broad classes (e.g., "joke", "movie", "star"), our approach is not superior after dataset filtering. This suggests that radicals may lose effectiveness when faced with comprehensive and modernized contents. Meanwhile, the higher performance of RAFG on class "constellation", "design" and "travel" shows its superiority on *dataset#2* compared with C-BLSTMs, which reveals that the effectiveness of radicals emerges when Chinese items are dominant in a text. Noticeably, there is a huge data losing in some classes (e.g., "digi", "lottery", "it", "design", "comic") on *dataset#2* when those non-Chinese items are filtered out, which decreases the average performance on *dataset#2*. However, compared with C-BLSTMs, RAFG achieves better performance on all these classes of *dataset#2* (i.e., "digi", "lottery", "it", "design", "comic"), which proves that RAFG can maintain robustness after introducing the information of radicals. In summary, the overall results on both *dataset#1* and *dataset#2* prove the rationality of RAFG and the importance of radical information when conducting Chinese text classification.

Conclusions

In this article, we presented an explorative study on Chinese text classification with a special focus on the utilization of radicals. During the exploration, we found that the word-level radicals, which are not usually noticed, have a good classification effect. What's more, we discovered that simply introducing radicals to Chinese text classification cannot improve the performance well. Inspired by these discoveries, we proposed our *Radical-Aware Attention-based Four-Granularity* (RAFG) model. Extensive experiments not only show the superiority of RAFG, but also validate the effectiveness of radicals in the task of Chinese text classification. In the future, we will explore the characteristics of Chinese and radicals in greater depth, since there are still many hidden patterns and rules to be discovered.

Acknowledgements

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Chen, X.; Xu, L.; Liu, Z.; Sun, M.; and Luan, H.-B. 2015. Joint learning of character and word embeddings. In *IJCAI*, 1236–1242.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 6645–6649. IEEE.

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

Hotho, A.; Nürnberger, A.; and Paaß, G. 2005. A brief survey of text mining. In *Ldv Forum*, volume 20, 19–62. Citeseer.

Huang, Z.; Liu, Q.; Chen, E.; Zhao, H.; Gao, M.; Wei, S.; Su, Y.; and Hu, G. 2017. Question difficulty prediction for reading problems in standard tests. In *AAAI*, 1352–1359.

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

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, 2267–2273.

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

Liu, W.; Allison, B.; Guthrie, D.; and Guthrie, L. 2007. Chinese text classification without automatic word segmentation. In *Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on*, 45–50. IEEE.

Liu, Q.; Wu, H.; Ye, Y.; Zhao, H.; Liu, C.; and Du, D. 2018. Patent litigation prediction: A convolutional tensor factorization approach. In *IJCAI*, 5052–5059.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751.

Peng, F.; Huang, X.; Schuurmans, D.; and Wang, S. 2003. Text classification in asian languages without word segmentation. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, 41–48. Association for Computational Linguistics.

Peng, H.; Cambria, E.; and Zou, X. 2017. Radical-based hierarchical embeddings for chinese sentiment analysis at sentence level. In *The 30th International FLAIRS conference. Marco Island*.

Peng, F.; Feng, F.; and McCallum, A. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*, 562. Association for Computational Linguistics.

Qiao, L.; Zhao, H.; Huang, X.; Li, K.; and Chen, E. 2019. A structure-enriched neural network for network embedding. *Expert Systems with Applications* 117:300–311.

Qin, C.; Zhu, H.; Xu, T.; Zhu, C.; Jiang, L.; Chen, E.; and Xiong, H. 2018. Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 25–34. ACM.

Shi, X.; Zhai, J.; Yang, X.; Xie, Z.; and Liu, C. 2015. 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)*, volume 2, 594–598.

Sun, Y.; Lin, L.; Yang, N.; Ji, Z.; and Wang, X. 2014. Radical-enhanced chinese character embedding. In *International Conference on Neural Information Processing*, 279–286. Springer.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. 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*, 1480–1489.

Yin, R.; Wang, Q.; Li, P.; Li, R.; and Wang, B. 2016. Multi-granularity chinese word embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 981–986.

Yu, J.; Jian, X.; Xin, H.; and Song, Y. 2017. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 286–291.

Zhou, Y.; Xu, B.; Xu, J.; Yang, L.; and Li, C. 2016. Compositional recurrent neural networks for chinese short text classification. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, 137–144. IEEE.

Zhou, X.; Wan, X.; and Xiao, J. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 247–256.

Zhu, X.; Sobihani, P.; and Guo, H. 2015. Long short-term memory over recursive structures. In *International Conference on Machine Learning*, 1604–1612.