International Journal of Innovative
Computing, Information and Control
Volume **15**, Number **1**, February **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **227–246**

CHARCNN-SVM FOR CHINESE TEXT DATASETS SENTIMENT CLASSIFICATION WITH DATA AUGMENTATION

Xingkai Wang[1,2], Yiqiang Sheng[1,*], Haojiang Deng[1] and Zhenyu Zhao[1,3]

[1]National Network New Media Engineering Research Center
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
{ wanxk; denghj }@dsp.ac.cn; *Corresponding author: shengyq@dsp.ac.cn

[2]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

[3]Department of Automation
University of Science and Technology of China
No. 96, Jinzhai Road, Baohe District, Hefei 230026, P. R. China
zhyzhao@mail.ustc.edu.cn

Received March 2018; revised July 2018

Abstract. *In recent years, social media has provided novel ways to gather information in significant quantities directly from users. Analyzing sentiment hidden in these data has proven useful for recommender systems, marketing, and monitoring public opinion. This paper demonstrates that using Chinese characters' unique tone as the input features improve accuracy in Chinese sentiment analysis. In terms of data augmentation, we construct a synonym replacement thesaurus for text classification. We propose a model that combines Char Convolutional Neural Networks (CharCNN) with a Support Vector Machine (SVM), which we call CharCNN-SVM, to obtain the emotional tendencies of users' reviews. Through experiments, results demonstrate that our method outperforms several traditional methods, such as Naive Bayes, maximum entropy, support vector machine and bag-of-words, in terms of Chinese sentiment analysis accuracy. We also note some useful phenomena and provide relevant explanations.*

Keywords: Sentiment classification, Char convolution neural network, Data augmentation, Support vector machine

1. **Introduction.**

With the development of social networks, an increasing number of people are expressing their opinions and feelings with others online. There are increasingly more mobile applications for providing users with these services. Users generate a large variety of comments when they use various mobile applications. The computational treatment of sentiment through public comments on social media has value in various business applications. For instance, through mining and analyzing the information of customer reviews on the Internet, companies can better understand the customers' emotional tendencies toward different enterprises and products, which could facilitate important business decisions for enterprises [1]. Analyzing people's opinions by extracting text from their reviews and comments has been proven useful for recommender systems [2]. Sentiment classification on short texts is currently an important topic in the emotional calculation field [3, 4, 5].

In the study of sentiment analysis, the most basic research is the construction of an emotional dictionary [6]. That is, text is processed by extracting emotional words based

DOI: 10.24507/ijicic.15.01.227

on the constructed emotional dictionary, and then the emotional tendency of the text is calculated. The effectiveness of the final classification depends on the integrity of the emotional dictionary. Another method is based on machine learning [7] by selecting emotional words as feature words. Then, logistic regression, naive Bayes, SVM, and other machine learning methods can be applied for classification. The effectiveness of the final classification depends on the choice of the training text and the correctness of the emotional annotations. It is also possible to combine two approaches [8]. In recent years, deep learning in natural language processing has provided effective results using an appropriate model. Researchers mainly study sentiment analysis at three levels of granularity: document level, sentence level, and aspect level [9]. This paper focuses on the document-level sentiment analysis problem. Document-level emotional categorization classifies documents (e.g., product reviews) with an overall positive or negative rating. It treats the entire document as a basic unit of information and assumes that documents contain the ideas from a single entity.

Most researchers pay little attention to the uniqueness of Chinese characters, such as tone. Our study, in contrast, is aimed at improving the classification accuracy of Chinese short text datasets by making good use of Chinese characters' uniqueness. We use Chinese texts' tone as input features because the tone is an expression of emotion. Each Chinese character is composed of vowels and consonants, and the tones are marked on the vowels. Processing Chinese text into the letter form will make it easier to extend our methods to many languages. In addition, some problems such as keyword sparseness exist in document classification. It is difficult to obtain features to represent documents due to a limited availability of keywords. This situation leads to low accuracy of text classification. To solve existing problems, we introduce a data augmentation method. Inspired by the success of combined Convolutional Neural Networks (CNN) and SVM in other fields, we propose a hybrid model, CharCNN-SVM, for Chinese sentiment analysis, which makes good use of the advantages of CharCNN and SVM. Previous experimental results showed that CharCNN can produce good results for Chinese text understanding without knowledge of the semantic structure [10]. As our experimental datasets are mainly Chinese user's short reviews, we apply CharCNN as an automatic feature extractor. A widely used CNN classification function is the Softmax at present. In this study, we focus on the binary classification of Chinese short text datasets. The reason that a consideration of the fusion of neural networks and SVM is the better performance for an SVM classifier over a Softmax classifier in binary classification problems [11]. Compared with the previous methods, our methods have improved the effectiveness of Chinese text dataset classification. The main contributions of this paper are as follows.

1) We propose a hybrid model based on CharCNN and SVM for Chinese sentiment analysis, which can better exploit the advantages of the CharCNN model's capabilities to extract deep features and the advantages of SVM in the binary classification problem. It is easy to apply this separate-model type to other models by separating the feature extraction module from the classifier module.
2) We introduce a data augmentation method to process datasets by constructing a synonym replacement thesaurus of the corresponding corpus. This method reduces the problem whereby a large number of low-frequency synonyms interfere with the text classification and solves the keyword sparseness problem. We construct attack datasets in Chinese sentiment analysis based on the data augmentation method, which is used to verify the model's robustness.
3) We note some useful phenomena in our experiments. The effectiveness of using Chinese language tones as input features was demonstrated. This improves the accuracy of the Chinese sentiment analysis. We also determined that the size of the

replacement thesaurus length should be relevant to the size of the datasets. The large size of the datasets requires a bigger replacement thesaurus length. Our experiments also show that the appropriate length of the model input frame should be relevant to the position where emotions are expressed in a sentence. We also provide relevant explanations based on the experimental results.

The remainder of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the data processing, character quantization, and CharCNN-SVM model in detail. Section 4 provides the evaluation methodology and experimental results. Section 5 presents the conclusions of our research and discusses future work.

2. **Related Work.** As early as the 1990s, foreign scholars began to study text sentiment analysis. Hatzivassiloglou and Mckeown extracted conjoined adjectives from large-scale corpus and analyzed the emotional polarity of these adjectives through a linear regression model. Finally, they clustered the adjectives using a clustering algorithm and verified from the corpus data constraints on the semantic orientations of conjoined adjectives [12]. Pang et al. examined the effectiveness of machine learning methods for sentiment classification tasks on movie reviews using three machine learning techniques, which were Naive Bayes, maximum entropy classification and SVM. SVM tended to provide the best performances [13]. Pang and Lee then proposed a method to deal with the results of semantic trend classification by subdividing the classification results according to the intensity of semantic orientation. A series of experiments proved the feasibility of the method [14]. In recent years, more and more scholars have studied how to apply deep learning to sentiment analysis. Deep learning using CNNs has achieved great success in visual recognition tasks such as image and text classification. Kim combined word embedding with convolutional networks in natural language tasks such as sentiment analysis and text categorization, and achieved favorable results [15]. Zhang and LeCun applied deep learning to text understanding from character-level inputs, progressing up to abstract text concepts [10]. At present, the study of Chinese sentiment analysis lags far behind studies abroad in quantity and depth. The method for English emotional classification cannot be used directly in the emotional classification of Chinese reviews. Zhu et al. proposed two methods of lexical semantic inclination calculation based on HowNet: a semantic similarity based method and a semantic related field based method [16]. The experiments showed that the methods were useful in Chinese common words. Ye et al. proposed a subjective automatic discrimination method for Chinese sentences based on 2-part-of-speech model, which provided an effective way to solve the problem in Chinese sentiment analysis [17]. Luo et al. reviewed the basic methods of social text normalization and discussed the future research directions of social text normalization [18].

In practical applications, some problems remain for Chinese sentiment analysis. One is extracting features from datasets. The effect of sentiment analysis is highly dependent on the design of features. It is universally acknowledged that linguists divide the world's languages into languages of tone and non-tone. Chinese is a tone language. Tone plays an important role in this language where four sounds are more complicated than in an accentual language. Chinese characters also contain more information than other languages. Based on these characteristics, we use Chinese language tones as input features. Chinese characters first need to be transformed into Pinyin, which is a phonetic system for transcribing Mandarin pronunciations. During this procedure, we used the Pinyin package combined with the jieba Chinese segmentation system.

Numerous scholars have found that data augmentation [19] techniques are useful for deep learning models to control generalization errors. Data augmentation has led to significant improvements in prediction accuracy and system robustness. However, text data augmentation is more difficult than image augmentation. In image processing, zooming and rotating will not change the meaning of an image. In text processing, we cannot arbitrarily replace the order of characters because their order represents the semantics. The authors in [20] proposed a data augmentation algorithm based on the theory of Gaussian random fields. They allowed the labels of the training cases to be propagated to the unlabeled data probabilistically. Text datasets always involve very high dimensions. A fact in high-dimensional space is that data tend to be orthogonal to each other. The authors in [21] represented documents with low-dimensional semantic-enriched features and augmented the training data with semi-supervised learning. Their algorithm was based on the graphical model of Gaussian random fields and its harmonic functions, and allowed the information from the labeled data to propagate to the unlabeled data based on the manifold of the combined data. We propose a method based on synonym extension instead. A simple synonym expansion is applied to detecting synonyms word-by-word in a small number of text datasets. If the word appears in the thesaurus, it will be extended or replaced. If the word appears in multiple rows of the thesaurus at the same time, the line with the highest frequency of appearances is used to expand the word. However, with this replacement, all synonym phrases always appear together in the text, which is apparently not realistic, as it is not conducive to text categorization. The author in [22] found that certain words are more useful in the corpus than other words, which we consider in our method. The researcher also hypothesized that simply adding duplicate words could lead to worse results. The expansion method based on a sentiment dictionary is actually similar to finding the synonym of the emotional word through the synonym dictionary. The basic emotional dictionary includes a number of widely recognized emotional words such as “good”, “beautiful”, “bad”, and “sad”. Some researchers have constructed emotional dictionaries, but these basic emotional dictionaries have many redundant or confused words for different linguistic datasets. The method of manual annotation involves manually selecting emotional words from special fields to reduce this problem, but involves significant time to accomplish. In this paper, we propose an automatic method for constructing a synonym replacement thesaurus. Expanding the corpus changed the datasets to a certain extent by using the dictionary we constructed. However, we accepted the change of text that this augmented method caused because it did not change the meaning of the text.

Significant attention has been paid to the fusion of neural networks and SVM. As mentioned above, it has been found that neural networks for natural language processing can be highly useful and SVM is popular for binary classification. The advantages of their combination have been confirmed in previous studies on intrusion detection in network traffic data [11], microvascular morphological type recognition [23], and handwritten digit recognition [24]. Our method is inspired by the successful combination of CNNs and SVM in other fields. As we mentioned above, CharCNN performs well in Chinese sentiment analysis and the SVM classifier performs better in binary classification problems than a Softmax classifier. In addition, the generalization capability of SVM is better than that of Softmax because the Softmax is based on empirical risk minimization, which attempts to minimize the prediction loss on the training set. In contrast, SVM aims to minimize the generalization error by applying structural risk minimization principles to the testing set [23]. Our experimental results demonstrated that the hybrid model effectively improved the accuracy and generalization capability for sentiment analysis in Chinese short text datasets.

3. CharCNN-SVM Model Design.

3.1. Motivation.

Our goal is to automatically learn the features of raw Chinese text datasets by using deep neural networks and to improve the accuracy of Chinese sentiment analysis. An implementation scheme is used for Chinese sentiment analysis, as shown in Figure 1. The scheme includes the stages of data processing, character quantization, features learning, and classification. First, data processing includes processing the datasets into Corpus A and Corpus B, data augmentation and converting Chinese characters to Pinyin. Then, we quantize characters as input to the CharCNN model. Next, every sentence is transformed into a two-dimensional image, the features of which are learned by the CharCNN model. Finally, the resulting features are used to determine whether their tendency is positive or negative by using SVM. We summarize the stages of our method in Algorithm 1. We discuss the design of our sentiment analysis method for Chinese text datasets from scratch in this section.

3.2. Data processing.

In this step, we segment the Chinese text datasets by using the jieba Chinese segmentation system. Then, an improved algorithm for text extension is proposed. Finally, we convert Chinese characters to Pinyin.

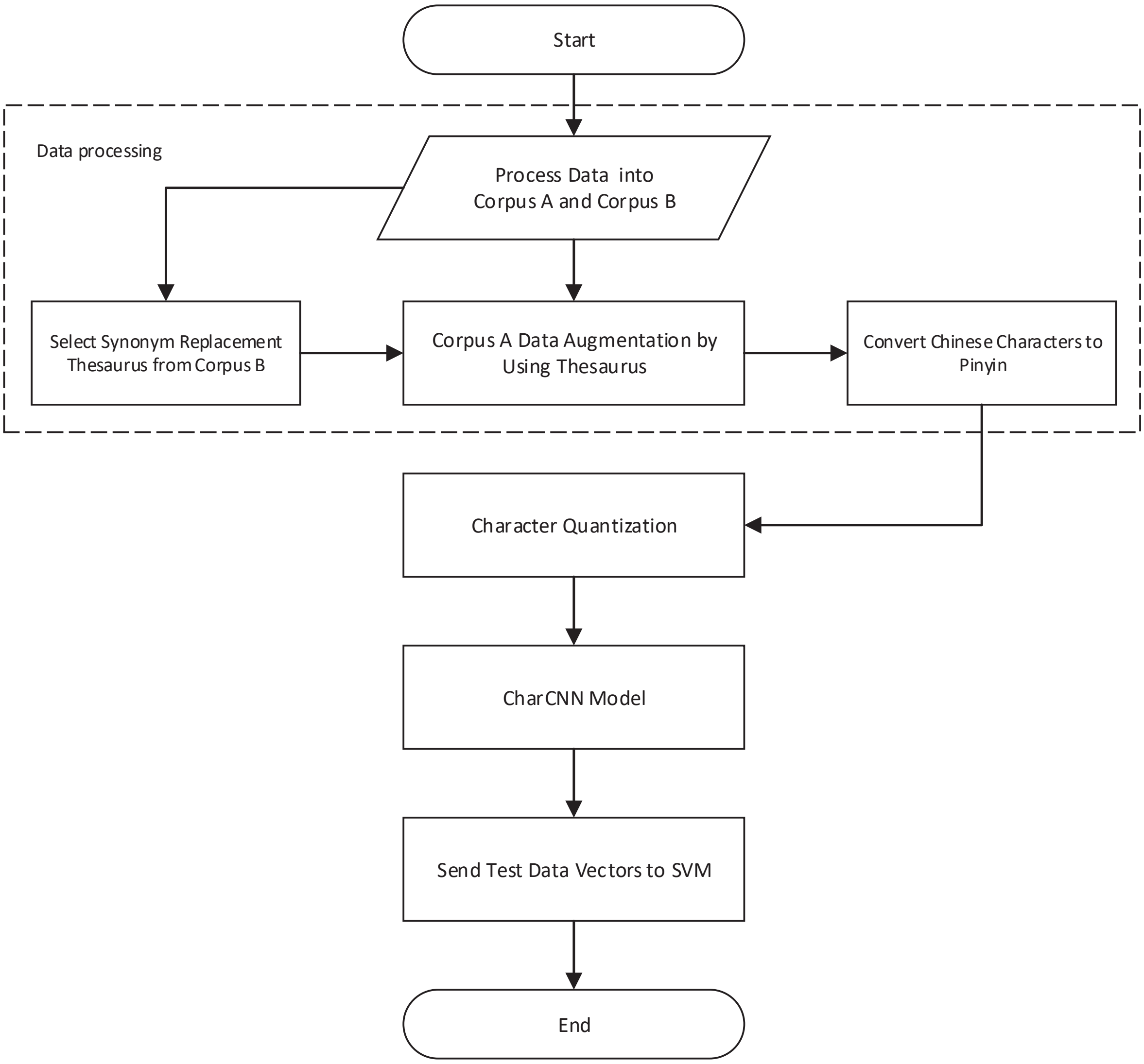
Start
Data processing
Process Data into Corpus A and Corpus B
Select Synonym Replacement Thesaurus from Corpus B
Corpus A Data Augmentation by Using Thesaurus
Convert Chinese Characters to Pinyin
Character Quantization
CharCNN Model
Send Test Data Vectors to SVM
End

Figure 1. Flow diagram of proposed system

Algorithm 1 Framework for our system

```
Input: raw Chinese text datasets
Output: the tendency of the sentence
// Data Processing
Process data into corpus A and corpus B;
Select synonym replacement thesaurus from corpus B;
Corpus A data augmentation by using thesaurus;
Convert Chinese characters to Pinyin;
//Character Quantization
Build two types of alphabets;
Quantize each character with 1-of-m encoding using the corresponding alphabet;
// Train and Validate Char Convolutional Neural Network Model
while early stop condition is not met do
    while training dataset is not empty do
        Prepare mini-batch dataset as the model input;
        Compute categorical cross-entropy loss function;
        Update weights and bias using Adam Optimization algorithm;
    end while
      Validate model using validating dataset;
end while
// Test model
Test model using test datasets;
Return the feature vectors of the full connection layer;
// Train and Test SVM Model
Send those vectors to the SVM model for classification;
Return the accuracy of tendency by using SVM.
```

3.2.1. *Word segmentation.* In this section, we use two methods to process the datasets into two forms by using the jieba Chinese segmentation system. For the first form we retain all the words and remove the punctuations, and label it as corpus A, which is fed to our model. In the second form, we retain the words containing semantic information by filtering all punctuations and meaningless special words such as time words, prepositions, and mimetic words, and label it as corpus B, from which we construct the synonym replacement thesaurus. Because some non-emotional words are useful to show different emotions in different contexts, we selected the synonym replacement thesaurus using corpus B rather than a corpus containing emotional words. For example, the words “hard” and “soft” represent a user’s emotions regarding hotel reviews, but they are not emotional words.

3.2.2. *Data augmentation using thesaurus.* At present, document classification has keyword sparseness problems. Therefore, how to expand the keywords is very important for improving the accuracy of text categorization. Simply adding words may not lead to better results. Instead, we augmented text with the constructed synonym replacement thesaurus, which is based on the “HIT IR-Lab Tongyici Cilin (Extended)” thesaurus. The vocabulary houses 12 major categories, 97 middle categories, 1400 sub-categories, and 77,343 words. Different words have different meanings in different contexts. For example, the words “hard” and “soft” can represent a user’s emotions in hotel reviews, but they do not represent a user’s emotions in laptop reviews. When these words do not express user emotions, they tend to be unrelated to classification in the current context. They should not be selected for the synonym replacement thesaurus. We used the chi-square statistics method to construct the thesaurus. In the stage of feature selection in text categorization, we mainly focus on whether a word and a category are independent of each other. Provided that they are independent, the word is not considered to characterize this category. That is, we cannot classify the text through the word. The chi-square statistics method is used to measure the correlation between two variables. Through the chi-square statistics method, we can judge whether a feature word is related to a certain classification. We assume that the word is unrelated to any categories as the original hypothesis, and then calculate the error between the actual relationship measure and the hypothesis. The larger the value of the calculation is, the greater the deviation from the original hypothesis is. The error is used to show correlations between words and categories. A greater error indicates that the feature word is more relevant to the classification. Calculating the chi-square value of a word t and category c is defined as Formula (1).

$$\chi^2\left(t,c\right) = \frac{N(AD-BC)^2}{\left(A+C\right)\left(A+B\right)\left(B+D\right)\left(C+D\right)} \tag{1}$$

Here, $N = A + B + C + D$ is the total number of documents, A is the number of documents that belongs to the category and contains the word, B is the number of documents that does not belong to that category but contains the word, C is the number of documents that belongs to the category but does not contain the word, and D is the number of documents that does not belong to the category and does not contain the word.

Of course, the chi-square statistics method is not perfect. It counts whether a word t appears in a document, regardless of how many times t appears in the document. This is the chi-square statistics method's limitation. Therefore, the chi-square statistics method is often combined with other measurements to avoid any weaknesses. Usual measurements in natural language processing are word count and Term Frequency-Inverse Document Frequency (TF-IDF) scores. However, our datasets involve various users' short comments in different scenarios. Very few words appear many times in the same text. Moreover, our method simply needs to select the most relevant terms in the document classification rather than extracting the features' weights. Using the chi-square statistics method is sufficient for our purposes. The specific implementation is as follows. First, we use the chi-square method to choose the synonym replacement thesaurus for expanding datasets rather than considering all the words' synonyms. We used the chi-square method to select the Top-N keywords from corpus B. Then, we formed a hash map collection by using the above keywords as the map's value. At the same time, we used the keywords' synonyms as the hash map's key. A keyword's corresponding synonym is believed to be semantically similar to it, and synonyms are obtained from the "HIT IR-Lab Tongyici Cilin (Extended)" thesaurus. The hash map is thus the synonym replacement thesaurus. Next, we experimented with two modification methods based on the thesaurus. The first is that if text contains the hash map collection's key, we use the corresponding value to replace the key. The second is that if text contains the hash map collection's key, we need to pick its corresponding value to add to the text. Finally, we found that the second method is superior to the first, so we used it to process our datasets. This process was repeated until all corpus A data were processed. We later report the results using this new data augmentation technique.

3.2.3. *Converting Chinese characters to Pinyin.* The final datasets form is Pinyin, which is a phonetic system for transcribing Mandarin pronunciations. During this procedure, we convert Chinese characters to Pinyin in two forms by using the Pinyin package. One form of the resulting Pinyin text has each tone appended on their letters, which includes characters such as “āáǎà”. The tone is an expression of emotion, so we use it as input. The other form of the resulting Pinyin text without tone consists of characters such as “abcd”.

3.3. **Character quantization.** The CharCNN model requires a sequence of encoded characters as input. We prescribed an alphabet of size m for the input language. Because we used two types of Pinyin text datasets, we built two alphabets for them. The alphabet for Pinyin text without tone consists of 37 characters, including 26 English letters and 10 digits:

Alphabet = “abcdefghijklmnopqrstuvwxyz0123456789”

At present, the tone symbols used in Chinese are: Yin Ping, Yang Ping, Shang Sheng, Qu Sheng, and tone symbols are added to the vowels. There are six Chinese vowels: a, e, i, o, u and v. However, vowel v does not have Yin Ping’s form in the Chinese dictionary, so the alphabet for Pinyin text with tone consists of 85 characters, including 26 English letters, 10 digits, 23 Chinese vowels, and 26 other characters:

Alphabet = “āáǎàabcdēéěèefghīíǐìijklmnōóǒòopqrstūúǔùuv́v̌v̀vwxyz0123456789”

We then quantize each character by using 1-of-m encoding. Next, these characters are transformed to m-sized vectors with fixed length l. If the length of the characters exceeds l, it will be ignored. If the length of the characters does not reach l, we use zeros to fill the character. In the experimental part, when we used the alphabet with tone to process the datasets without tone, we achieved poor results. When we used the alphabet without tone to process the datasets with tone, we also achieved poor results. However, when we used the corresponding alphabet to process the corresponding corpus, we achieved good results. A possible explanation is that any characters not in the alphabet are quantized as zero vectors, which introduces useless information. Our experimental results proved the effectiveness of adding tones as input features. In addition, as CharCNN can learn from simple quantization, we fed the input to our model without other normalization.

3.4. **Proposed hybrid system.** The architecture of the hybrid model is shown in Figure 2. To utilize the CharCNN, we built the alphabets. We processed Chinese text into characters and quantized the characters as 1-of-m encoding using the alphabet. Instead of using the more complicated structure of neural networks, we adopted a simplified CNN structure whose architectural parameters are presented in Section 4.1.3. We used the CharCNN to extract deep features from the Chinese text, which are represented by high-dimensional vectors. Then, we send these high-dimensional vectors to the SVM. Finally, the SVM completed the Chinese sentiment classification. We will briefly introduce the CharCNN structure in Section 3.4.1, the SVM theory in Section 3.4.2, and the hybrid CharCNN-SVM model in Section 3.4.3.

3.4.1. *Char convolutional neural networks.* In this section, we discuss the CharCNN model used to extract features. Compared with other feature extraction methods, it is easier to extend to many human languages. The network is normally trained using gradient backpropagation [25] techniques to perform optimization. In the experiment, we used different convolution, pooling, and the size of the final output eigenvectors to control the overall model fitting capability. The architecture of CharCNN is shown in Figure 3. We built two alphabets for two types of Pinyin text datasets, so the input vectors have a number of frames equal to the length of each alphabet.

Convolution kernels extract multiple sets of local features. A convolution operation involves a filter $w \in R^{(h*k)}$. Here, “h” is the number of vertical characters and “k” is the

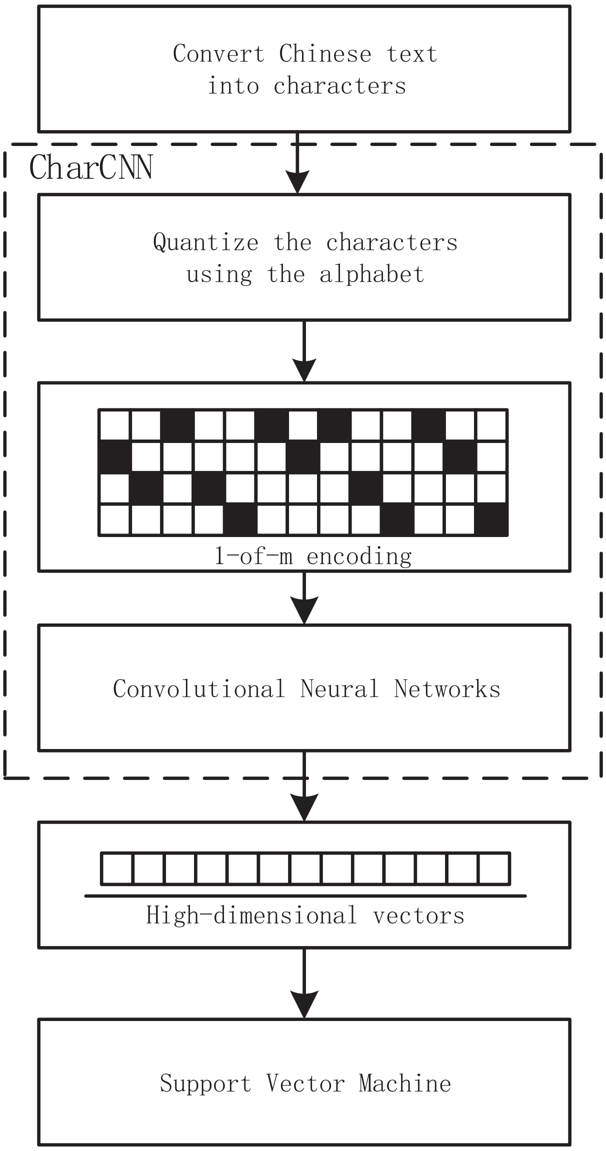

Figure 2. Illustration of CharCNN-SVM model

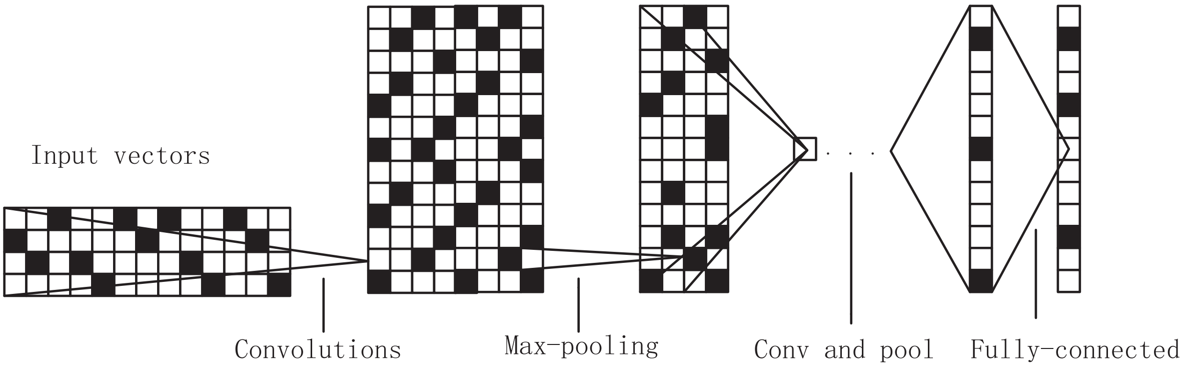

Figure 3. Illustration of CharCNN model. It was inspired by Zhang and LeCun's model [10].

dimension of a character vector. Formula (2) indicates how feature S_i is generated from a window of characters $a_{i:i+h-1}$.

$$S_i = f(w * a_{i:i+h-1} + b) \tag{2}$$

of which b is the offset and f is the activation function.

From this function, we can obtain a feature map S $\left(S \in R^{(m-h+1)}\right)$ through one filter for convolution operations. The notation m refers to the number of characters in every sentence. As the representative word in the language could reflect the emotion tendency of each sentence, max-pooling is used to complete the process of extracting the local optimal features. Formula (3) describes the process.

$$f(x) = \max(0, x) \tag{3}$$

Then, fully connected layers are used to complete the emotional mapping, and dropout layers are added in the fully connected layers for regularization.

3.4.2. *Support vector machine.* The support vector machine [26] is used as the final classifier in our proposed system. SVM with different kernel functions can transform a nonlinear separable problem into a linear separable problem by transforming the feature vectors into high-dimensional space to locate the optimal separate hyperplane. Given a training set of instance label pairs (x_i, y_i), $i = 1, \ldots, n$, where $x_i \in R^n$ and $y \in (1, -1)$, the SVM requires the solution to the following optimization problem, which is described in Formula (4).

$$\min_{w,b,\varepsilon} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \varepsilon_i, \quad \text{subject to} \quad y_i \left(w^T \phi(x_i) + b\right) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0 \tag{4}$$

Here, w is the weight vector for x, C is an adjustable penalty parameter controlling the trade-off between the maximization of margin and the minimization of classification errors, b is a scalar, ε_i are the slack variables, vectors x_i are mapped into higher dimensional space by the function ϕ, and $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is called the kernel function [27].

The SVM classifier was originally proposed for binary classification. It has characteristics such as a convex loss function and a maximum margin criterion that provide good generalization capabilities. In this paper, we use LIBSVM [28] as the SVM tool, which is an effective SVM library for classification problems. Four popular kernels are provided in LIBSVM [27]. The Radial Basis Function (RBF) kernel is a reasonable first choice, which is described by Formula (5).

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \quad \gamma > 0 \tag{5}$$

of which γ is a kernel parameter.

The performance of the RBF kernel depends on the setting of its parameters (C, γ). In LIBSVM, it is recommended to conduct a "grid-search" on C and γ using cross-validation. In our experiments, we determined the initial parameters (C, γ) using the LIBSVM interface – grid.py.

3.4.3. *Hybrid CharCNN-SVM model.* In current CNNs, the Softmax function has been used as a popular function for the final layer. The Softmax is good at solving a multi-class classification problem. However, it must perform all calculations again to satisfy a probability distribution when a new class is added into classification problem. In this paper, we focus on the binary classification of Chinese sentiment analysis, so SVM, known to perform better than Softmax in binary classification, is chosen. Softmax requires a probability distribution iterating through a K-dimensional vector, while the SVM only has to find the optimal hyperplane. That is, SVM is a more reasonable approach than Softmax for binary classification. Despite SVM having some attractive characteristics such as a convex loss function and a maximum margin criterion, it has some limitations in text classification. SVM is a shallow architecture; therefore, it presents some difficulties for learning deeper hierarchical features. However, the CharCNN model is built by using a hierarchical architecture that is able to learn deeper hierarchical features presented by the Chinese text datasets. The architecture of our hybrid model was designed by separating the feature extraction module from the classifier module. That is to say, the CharCNN acted as a feature extractor and the SVM acted as a predictor. The CharCNN extracts deep features in the form of high-dimensional vectors. These outputs from the hidden layer with trainable weights and a bias make sense, as they can generate estimated probabilities for the input sample by using the CharCNN's classifier – a Softmax classifier. Meanwhile, these outputs can be treated as features for other classifiers. Through the discussion above, we used the SVM with RBF kernels to take these outputs from the CharCNN's hidden layer as feature vectors for training instead of Softmax. Thus, the SVM model completed the final classification in Chinese sentiment analysis. We effectively separated the feature extraction module from the classifier module, which could better use the advantages of the CharCNN model to extract deep features and the advantages of SVM for binary classification. It is also easy to apply this separate model concept to other models.

In this paper, we designed an effective system to classify Chinese text corpus. The model makes good use of the ability to extract features from CharCNN model and takes advantage of the excellent performance of SVM for binary classification. In our experiments, when the CharCNN model converged and the accuracy of the validating datasets achieved the best value, the test output vectors of the final fully connected layer were transformed to the format required by LIBSVM.

We extracted feature words that are most relevant to document classification in different corpora based on the basic chi-square (Formula (1)), and this process step is only performed once for different corpora. In addition, because the length of the synonym replacement thesaurus is limited, the complexity of the replacement process is $O(n)$. After data augmentation, we used the CharCNN-SVM model to classify the datasets. As we do not add high complexity algorithms in our method, our method is effective in terms of complexity. Our experimental results demonstrate that the hybrid model achieved by combining the good attributes of CharCNN and SVM is beneficial in Chinese sentiment analysis.

4. **Evaluation and Discussion.** This section evaluates the performance of our method via various experiments on different datasets. More specifically, our results aim to achieve the following:

1) Evaluate the effectiveness of synonym replacement thesaurus length size on the performance of our method.
2) Evaluate the effectiveness of input frame length size on the performance of our method.
3) Show the best experimental results of our method and make a comparison with other models.

4.1. **Experimental methodology.**

4.1.1. *Datasets.* The unfortunate fact in Chinese text datasets classification is that there is no openly accessible dataset that is large enough or has labels of sufficient quality for us, although research on Chinese text understanding has been conducted for decades. We adopted several of the current open Chinese emotion mining corpus – ChnSentiCorp [29] as experimental datasets, including Hotel Reviews Corpus, Laptop Reviews Corpus, and Dangdang Book Reviews.

Hotel Reviews Corpus. Among the hotel reviews, approximately 10000 texts contain both emotional polarity and some contents. Of these, 7000 are positive, and the others are negative. The reviews were collected from Ctrip online and were divided into two subsets. From the full dataset, we randomly selected 8000 samples for training, 1000 samples for validation, and 1000 samples for testing.

Laptop Reviews Corpus. The reviews were collected from Jingdong, which is an electronic shopping platform. It has two subsets. One has 2000 texts containing both positive emotional polarity and some contents. The other 2000 texts include negative emotional polarity and some contents. From each subset, we randomly selected 1800 samples for training, 100 samples for validation, and 100 samples for testing.

Dangdang Book Reviews. These reviews were collected from Dangdang, which is an electronic book platform. There are 4000 texts that contain both emotional polarity and some contents in this dataset. 2000 of them are positive, and the others are negative. In this dataset, the total size of the training dataset is 3600, validation dataset 200, and testing dataset 200.

4.1.2. *Metrics.* Two metrics were used to evaluate the performance of our methods: accuracy and robustness. To compute the accuracy, we obtain the number of correct_predictions, defined as Formula (6).

$$correct\_prediction\,(y, y^*) = \begin{cases} 1, & y = y^* \\ 0, & y \neq y^* \end{cases} \tag{6}$$

where y is the model's total predicted labels and y^* is the number of actual labels. The calculation formula for *accuracy* can be expressed as Formula (7).

$$accuracy = \frac{\sum_{i=1}^{n} correct\_prediction_i}{n} \tag{7}$$

where, n is the number of samples. Accuracy was used to evaluate the overall performance of our method. Compared to those of previous methods, this paper's experimental results fully prove the high efficiency of our proposed methods regarding accuracy improvement. As mentioned above, SVM as the classifier provides good generalization capabilities. We measure the robustness of the model by the defined value *robustness.* The formulation for measuring the *robustness* is given in Formula (8).

$$robustness = \frac{acc - acc^*}{acc} \tag{8}$$

In the above formula, acc is the accuracy of normal samples and acc^* is the accuracy of attack samples. We can determine the influence of an attack sample on the model by comparing the size of the *robustness.* The lower the value of *robustness* is, the stronger the model's disturbance resistant capability is. The experimental results show that our hybrid model is more robust to attack samples than the basic CharCNN model.

4.1.3. *Experimental setup.* Tables 1, 2 and 3 describe the architectures of the CharCNN. We designed three ConvNets, which included convolution layers, pooling layers, and fully connected layers. These layers have different numbers of hidden units and frame sizes for different datasets. All implementation schemes contain a Softmax layer as the classifier. Architecture parameters were fine-tuned after several experiments were conducted.

Our Char Convolutional Neural Network was built with tensorflow [30], a deep learning software framework run on the Ubuntu 16.04 64 bit OS. The server is a desktop computer with AMD RYZEN7 CPU and 16 GB of memory. An Nvidia 1060 GPU with 6 GB of

Table 1. DNN architecture parameters for hotel reviews

Layer	Type	Neuron/Kernel	Pool	Pad
1	*Conv* + *ReLU*	128/7	3	same
2	*Conv* + *ReLU*	128/7	None	same
3	*Conv* + *ReLU*	128/3	3	same
4	dense	512	–	none
5	dense	512	–	none
6	dense	2	–	none

Table 2. DNN architecture parameters for laptop reviews

Layer	Type	Neuron/Kernel	Pool	Pad
1	*Conv* + *ReLU*	128/7	3	same
2	*Conv* + *ReLU*	128/3	3	same
3	dense	512	–	none
4	dense	2	–	none

Table 3. DNN architecture parameters for book reviews

Layer	Type	Neuron/Kernel	Pool	Pad
1	*Conv* + *ReLU*	128/7	3	same
2	*Conv* + *ReLU*	128/3	3	same
3	dense	512	–	none
4	dense	2	–	none

Table 4. Training hyper-parameters of our model

Hyper-parameter	Hotel reviews	Laptop reviews	Book reviews
Dropout rate	0.5	0.5	0.5
Loss function	Cross-entropy	Cross-entropy	Cross-entropy
Optimizer	Adam	Adam	Adam
Mini-batch size	64	32	64
Learning rage	le-4	le-4	le-4

memory was used as the accelerator. The training hyper-parameters are shown in Table 4.

4.2. **Influence of synonym replacement thesaurus.** We propose a method that constructs the synonym replacement thesaurus of the corresponding corpus for data augmentation. We use the chi-square statistics method to select the Top-N keywords from those in corpus B. Every length of the synonym replacement thesaurus should be selected to fit a pre-defined size in the processing stage for the different datasets. The numbers of words in the different datasets varied. Table 5 shows the word counts for the three datasets.

As shown in Table 5, the numbers of words in the three datasets have large differences. A method for choosing the suitable length of the synonym replacement thesaurus needs to be determined via experiments. Figure 4 shows the experimental results of our method for the three datasets regarding the length of the synonym replacement thesaurus. In our experiments, the length size ranges from 100 to 1000. When the size is larger than 600, the performance no longer improves in hotel reviews. When the size is larger than 200, the performance no longer improves in laptop reviews. Finally, when the size is larger

TABLE 5. Word counts

Word count	Hotel reviews	Laptop reviews	Book reviews
Number	693275	131207	272965

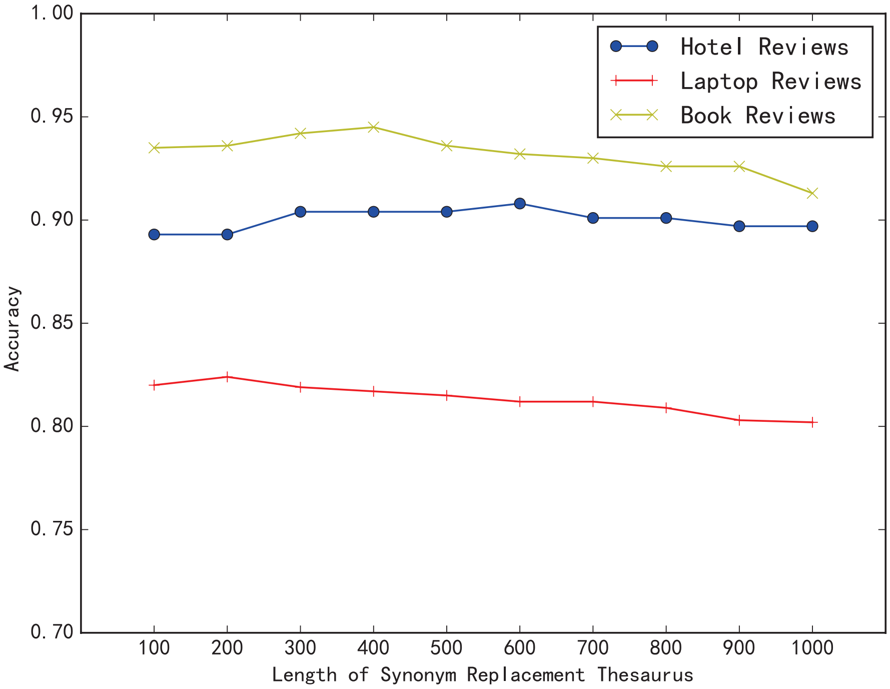

FIGURE 4. Results for different lengths of synonym replacement thesaurus

than 400, the performance no longer improves in book reviews. Note that the suitable length of synonym replacement thesauruses is different. A possible explanation is that there are as many keywords associated with classification in datasets with a large number of words as for those with a small number of words. So the appropriate size of thesaurus should be larger. The results can also prove that when the size reaches a certain limit, the performance no longer improves. Some words appear in multiple rows in the "HIT IR-Lab Tongyici Cilin (Extended)" thesaurus at the same time because there are several different meanings for those words in Chinese. So a possible explanation is that irrelevant words are less useful for expanding the corpus than relevant words. Adding many irrelevant words resulted in worse experimental results. Note that laptop reviews and book reviews contain the same number of samples, but their accuracy curves vary significantly. So, we analyzed the contents of these texts and also analyzed the high-frequency words and low-frequency words in the different corpora. We found that laptop reviews lacked normative. These texts contain many abbreviations, spoken language, and popular online language. Because most of the users who buy laptops online are young people, the comment style of their reviews is more novel. For example, a negative comment in laptop reviews is that the plastic of the optical drive is like a pancake. So the laptop reviews' low-frequency words had a larger amount of information than others. At the same time, these words may represent interference factors. In contrast, the comments in book reviews are more standardized. This is the reason that the accuracy of book reviews is higher in our experiments, where the evaluation metric is accuracy of text classification.

4.3. **Influence of the length of frames.** Every sentence must be reduced to a fixed size, which is a special requirement of the CNN for input data. In the field of sentiment analysis, the length of sentences is not fixed. Table 6 shows some statistics from the three datasets. Large differences in length of sentences exist in all three datasets. We concatenated emotional polarity and content to form input samples. As mentioned above, these sentences were transformed to m-sized vectors with fixed length l. If the length of the characters exceeded l, it was ignored. If the length of the character did not reach l, we use zeros to fill the character. Of course, the zeros will introduce more useless information. Because the settings of our neural networks and the lengths of most texts in our datasets were less than 1000, the length size ranged from 200 to 1000. A method for choosing a suitable length needs to be determined via experiments. Figure 5 shows the experimental results of our method for the three datasets regarding the length of frames. In our experiments, when the length reached 500, accuracy improvement was not obvious in hotel reviews. When the length reached 300, accuracy improvement was not obvious in laptop reviews. Finally, when the length reached 500, accuracy improvement was not obvious in book reviews. Note that the suitable length has an important effect on the model's performance. A possible explanation is that quite a few emotions are expressed at the beginning of a sentence. The comment style of laptop reviews is more concise, so the appropriate frame length is shorter than for the others. Considering the training time, we chose the shorter lengths as the most appropriate on the basis of optimal accuracy. As stated previously, the evaluation metric is accuracy of text classification.

Table 6. Statistics of datasets

Length of sentence	Hotel reviews	Laptop reviews	Book reviews
Max	7611	532	1179
Min	1	25	6
Mean	341.4	157	321
Median	222	135	353

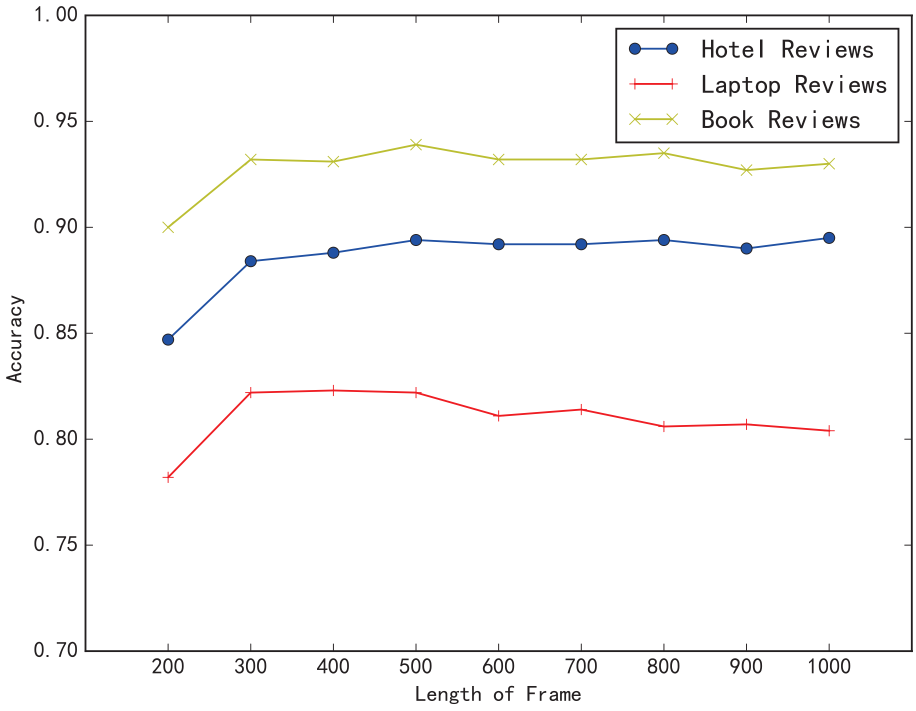

Figure 5. Results for different lengths of frames

4.4. **Comparison with other published methods.** This section shows the best experimental results of our method and makes a comparison with other models. In this section, we implement several standard models by using previous methods: the bag-of-words model, Naive Bayesian model, maximum entropy model, a method based on SVM, and the basic CharCNN.

The bag-of-words model is a simplified expression model used in natural language processing and information retrieval. In this paper, we counted how many times each word appears in the training dataset, and chose the most frequent appropriate number of words as the bag in different datasets. We created features from the bag by using the scikit-learn [31] package. Finally, we used logistic regression as the classifier for this bag of features.

We implemented the Naive Bayesian and maximum entropy models by using MALLET [32]. MALLET is a Java-based package for document classification and other machine learning applications involving text. Our source data consists of many separate files. MALLET provides a method for our source data type by importing data into MALLET format. In this paper, we chose Naive Bayes and maximum entropy classifiers to complete document classification using MALLET.

The method based on SVM includes the following steps: text word segmentation, feature selection, feature weight calculation, text vector representation, model training, and model prediction. In this part of the experiment, we used LIBSVM as the SVM tool.

As mentioned above, we selected the most suitable parameters of our models for the three datasets. Figures 6, 7 and 8 list the results. The numbers provide best accuracy in classification.

The above figures show that using tone as input achieved better performances regarding the accuracy of Chinese sentiment analysis. Our data augmentation method also improved the accuracy of classification. In addition, the experimental accuracy of CharCNN-SVM was higher than the experimental accuracy of the CharCNN model. As mentioned in Section 3.4.3, the superiority of the CharCNN model is in automatically extracting the features of samples, while the SVM classifier is better in binary classification. So combining the merits of CharCNN and SVM is beneficial for enhancing the classification accuracy.

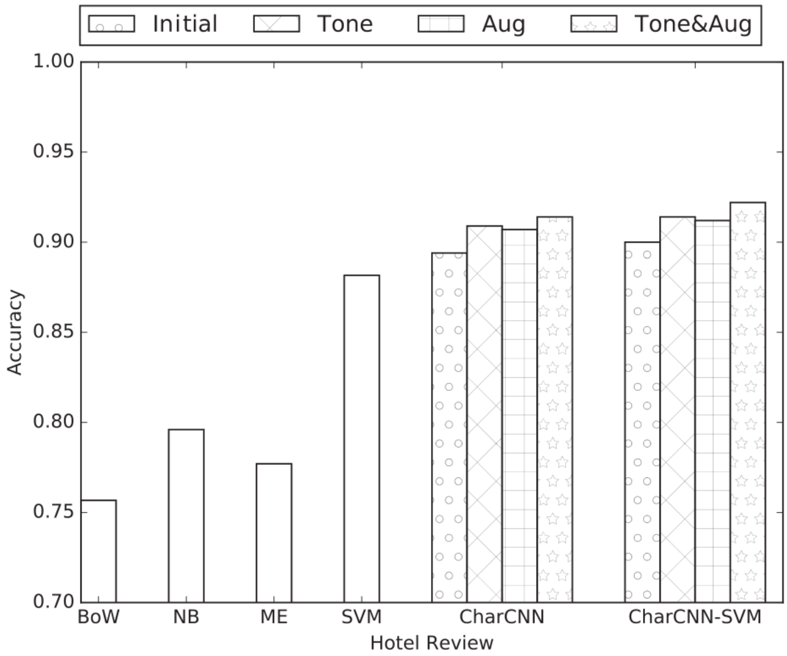

Figure 6. Results for hotel reviews

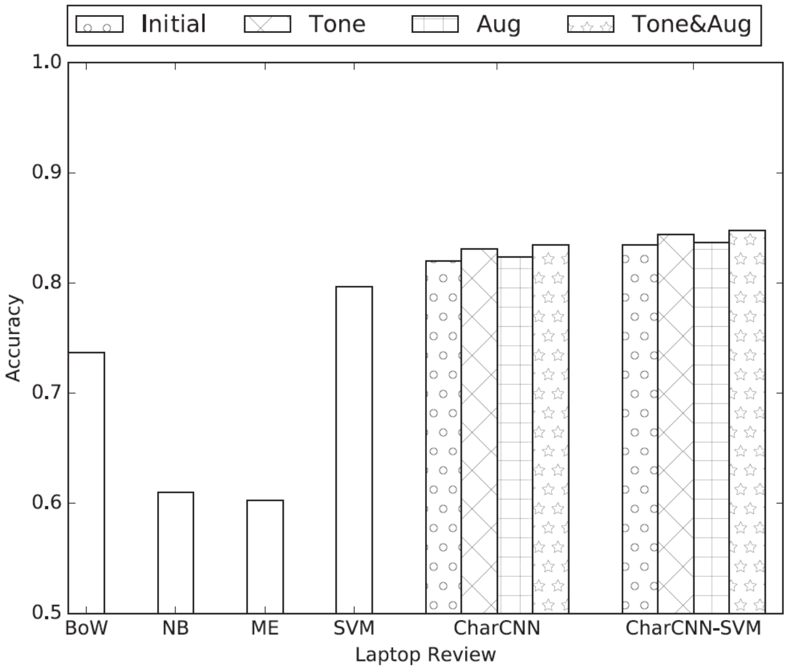

Figure 7. Results for laptop reviews

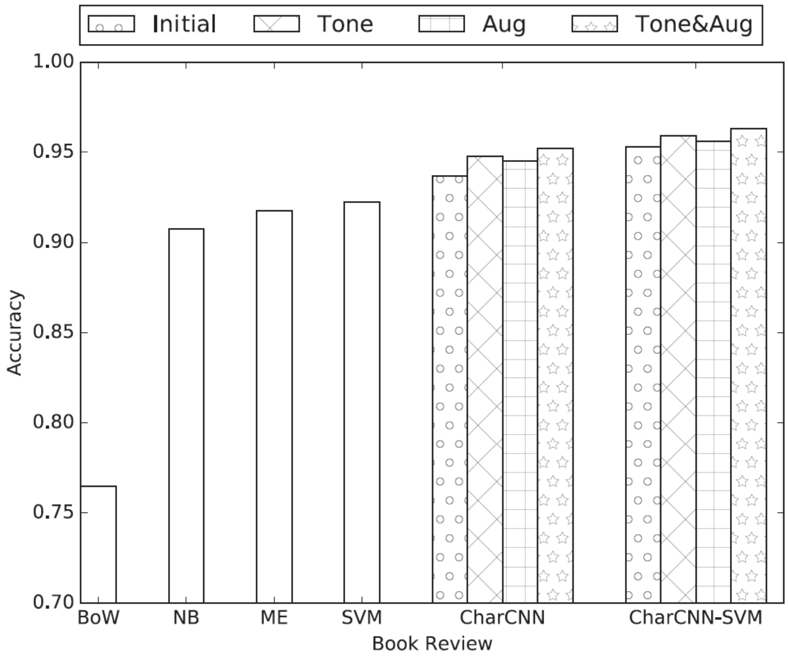

Figure 8. Results for Dangdang book reviews

However, the improvement of CharCNN-SVM over CharCNN is not significantly obvious. One explanation is that the Chinese text training samples are not large enough. The hybrid model requires more training samples to achieve better accuracy. Rich data could improve the model's classification capabilities.

Our experiments demonstrate that our method improves the accuracy of sentiment analysis in Chinese text classification. It outperforms the above traditional methods in terms of accuracy, so we believe that applying our method to a more effective deep learning model could produce a better effect.

4.5. **Verifying robustness.** Our hybrid model has strong disturbance resistance, owing to the advantage of SVM. We proved it by constructing attack samples. We used the "HIT IR-Lab Tongyici Cilin (Extended)" thesaurus and Chinese emotion vocabulary ontology data to construct attack datasets. The Chinese emotion vocabulary ontology library [33] is a Chinese ontology resource which is ordered and marked by the efforts of all the teaching staff members under the guidance of Professor Lin Hongfei in the Information Retrieval Laboratory of Dalian University of Technology. The resource describes a Chinese vocabulary or phrase from different perspectives, including the type of emotion, the intensity of emotion, and the polarity.

First, we formed a new hash map collection. We obtained the intensity of emotion of synonym replacement thesaurus key from the Chinese emotion vocabulary ontology. Then, we obtained the intensity of the emotion corresponding value. If the key's emotional strength is less than the value's, we consider the synonym replacement thesaurus key as the new hash map collection's value. Then, we assign the corresponding value as the new hash map collection's key. Finally, if corpus A's text contains the new map collection's key, we replace the key with the corresponding value in the text. This method reduces the emotional intensity of the data set. We accepted this change because it did not change the meaning of text.

We used the above method to generate the hotel datasets attack sample. Then we inputted it to the CharCNN model after selecting the most suitable parameters for it. Next, when the CharCNN model converged, we sent the test output vectors of the final fully connected layer that were transformed to the format required by LIBSVM to the SVM model for classification. Figure 9 shows the results of the comparison between the two models. The numbers represent the accuracy of classification. We determined that $robustness(CharCNN) = 0.023$ and $robustness(CharCNN\text{-}SVM) = 0.010$. Our results show that the CharCNN-SVM model appears to be beneficial regarding generalization capability in Chinese sentiment analysis. As we mentioned in Section 2, the SVM classifier uses the structural risk minimization principle to minimize the generalization errors on training datasets, which leads to a generalization capability higher than that of

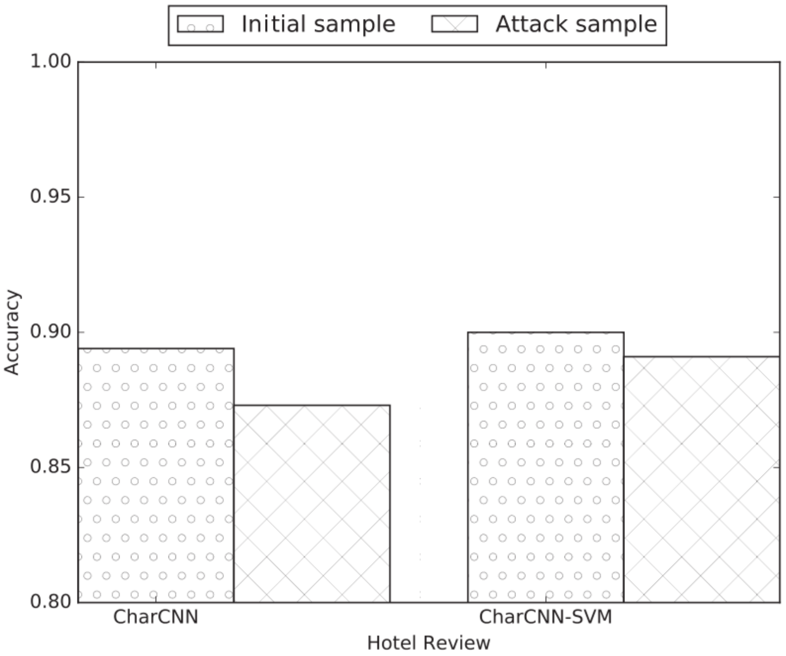

FIGURE 9. Attack data results on Song-bo tan hotel reviews corpora

the Softmax classifier. So the generalization capability of SVM plays a role in improving the CharCNN-SVM's robustness.

5. **Conclusion and Future Work.** In this paper, a hybrid CharCNN-SVM model has been proposed for Chinese text understanding tasks. Using the Chinese corpus's four tones annotation as input characteristics is innovative. Our data augmentation method not only is simple but also reduces secondary interference, which involves low-frequency synonyms appearing in the text. In addition, we also noted some useful phenomena and provided relevant explanations based on our experimental results.

Our preliminary experimental results are very promising. Good results were achieved on Chinese sentiment analysis by feature selection, data augmentation, and a hybrid CharCNN-SVM model. Experimental results showed that our method performed better than several previous methods in terms of accuracy and proved that the hybrid model is more robust.

There are opportunities to improve our method. The first involves solving the cross-domain sentiment analysis problem. Transfer learning is one of the main research areas in this field. It is also worth noting that time-series data could provide improvements in sentiment analysis tasks. We will focus on these areas in the future.

Acknowledgment. This work is supported by the Special Fund for Strategic Pilot Technology of Chinese Academy of Sciences under Grant No. XDA06040602.

REFERENCES

[1] M. van Zaanen and L. M. Dewi, Meaning of sentiment analysis for companies, *The 3rd PIABC (Parahyangan International Accounting and Business Conference)*, 2017.

[2] G. Preethi, P. V. Krishna, M. S. Obaidat, V. Saritha and S. Yenduri, Application of deep learning to sentiment analysis for recommender system on cloud, *International Conference on Computer, Information and Telecommunication Systems*, pp.93-97, 2017.

[3] S. Kiritchenko, X. Zhu and S. M. Mohammad, Sentiment analysis of short informal texts, *Journal of Artificial Intelligence Research*, vol.50, pp.723-762, 2014.

[4] X. Zhang, H. Li and L. Wang, A context-based regularization method for short-text sentiment analysis, *International Conference on Service Systems and Service Management (ICSSSM)*, pp.1-6, 2017.

[5] C. dos Santos and M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts, *Proc. of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp.69-78, 2014.

[6] Y. Rao, J. Lei, W. Liu, Q. Li and M. Chen, Building emotional dictionary for sentiment analysis of online news, *World Wide Web – Internet Web Information Systems*, vol.17, no.4, pp.723-742, 2014.

[7] E. Boiy and M. F. Moens, A machine learning approach to sentiment analysis in multilingual web texts, *Information Retrieval*, vol.12, no.5, pp.526-558, 2009.

[8] G. Gaikwad and D. J. Joshi, Multiclass mood classification on Twitter using lexicon dictionary and machine learning algorithms, *International Conference on Inventive Computation Technologies*, pp.1-6, 2017.

[9] L. W. Ku, Y. T. Liang and H. H. Chen, Opinion extraction, summarization and tracking in news and Blog corpora, *Proc. of AAAI*, pp.100-107, 2006.

[10] X. Zhang and Y. LeCun, Text understanding from scratch, *arXiv preprint arXiv:1502.01710*, 2015.

[11] A. F. Agarap, A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data, *arXiv:1709.03082*, 2018.

[12] V. Hatzivassiloglou and K. R. Mckeown, Predicting the semantic orientation of adjectives, *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pp.174-181, 2009.

[13] B. Pang, L. Lee and S. Vaithyanathan, Thumbs up?: Sentiment classification using machine learning techniques, *Proc. of the ACL-02 conference on Empirical Methods in Natural Language Processing*, vol.10, pp.79-86, 2002.

[14] B. Pang and L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, pp.115-124, 2005.

[15] Y. Kim, Convolutional neural networks for sentence classification, *arXiv preprint arXiv:1408.5882*, 2014.

[16] Y. Zhu, J. Min and Y. Zhou, Semantic orientation computing based on HowNet, *Journal of Chinese Information Processing*, vol.20, no.1, pp.14-20, 2006.

[17] Q. Ye, Z. Zhang and Z. Luo, A study on the automatic chinese subjectivity discrimination based on emotional analysis of internet comments, *China Journal of Information Systems*, vol.1, no.1, 2007.

[18] C. Luo, X. Wu, K. Xue, F. Yang and B. Wang, An overview of social text normalization, *Journal of Network New Media*, no.5, pp.10-14, 2017.

[19] K. Chatfield, K. Simonyan, A. Vedaldi and A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, *arXiv preprint arXiv:1405.3531*, 2014.

[20] X. Zhu, Z. Ghahramani and J. D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, *The 20th International Conference on Machine Learning*, pp.912-919, 2003.

[21] X. Lu, B. Zheng, A Velivelli and C. Zhai, Enhancing text categorization with semantic-enriched representation and training data augmentation, *Journal of the American Medical Informatics Association*, vol.13, no.5, p.526-535, 2006.

[22] J. Ramos, Using TF-IDF to determine word relevance in document queries, *Proc. of the 1st Instructional Conference on Machine Learning*, 2003.

[23] D.-X. Xue, R. Zhang, H. Feng and Y.-L. Wang, CNN-SVM for microvascular morphological type recognition with data augmentation, *Journal of Medical and Biological Engineering*, vol.36, no.6, pp.755-764, 2016.

[24] X.-X. Niu and C. Y. Suen, A novel hybrid CNN-SVM classifier for recognizing handwritten digits, *Pattern Recognition*, vol.45, no.4, pp.1318-1325, 2012.

[25] D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning representations by back-propagating errors, *Nature*, vol.323, 1986.

[26] C. Cortes and V. Vapnik, Support vector machine, *Machine Learning*, vol.20, no.3, pp.273-297, 1995.

[27] C.-W. Hsu, C.-C. Chang, C.-J. Lin et al., *A Practical Guide to Support Vector Classification*, 2003.

[28] C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, *ACM Trans. Intelligent Systems and Technology (TIST)*, vol.2, no.3, 2011.

[29] S. Tan, Chinese emotion mining corpus – ChnSentiCorp resources from nlpir.org, *Natural Language Processing and Information Retrieval Sharing Platform*, 2012.

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean and M. Devin, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, *arXiv:1603.04467*, 2016.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, Scikit-learn: Machine learning in python, *Journal of Machine Learning Research*, vol.12, pp.2825-2830, 2011.

[32] A. K. McCallum, *Mallet: A Machine Learning for Language Toolkit*, http://mallet.cs.umass.edu, 2002.

[33] L. Xu, H. Lin, Y. Pan, H. Ren and J. Chen, Constructing the affective lexicon ontology, *Journal of the China Society for Scientific and Technical Information*, vol.27, no.2, pp.180-185, 2008.