Received: 6 July 2019 | Revised: 6 November 2019 | Accepted: 18 November 2019
DOI: 10.1002/cpe.5627

WILEY

SPECIAL ISSUE PAPER

Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts

Nijia Lu[1] | Guohua Wu[1] | Zhen Zhang[1] | Yitao Zheng[1] | Yizhi Ren[1] | Kim-Kwang Raymond Choo[2]

[1]School of Cyberspace, Hangzhou Dianzi University, Zhejiang, China
[2]Department of Information Systems and Cyber Security and Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, Texas

Correspondence
Yizhi Ren, School of Cyberspace, Hangzhou Dianzi University, Zhejiang, China.
Email: renyz@hdu.edu.cn

Present Address
Zhen Zhang, 1158, No. 2 Street, Baiyang Street Hangzhou, Zhejiang, CN 310018

Funding information
Zhejiang Province Natural Science Foundation, Grant/Award Number: No. LY18F020017; National Natural Science Foundation of China, Grant/Award Number: No. 61872120

Summary

As people spend increasingly more time on social networks, cyberbullying has become a social problem that needs to be solved by machine learning methods. Our research focuses on textual cyberbullying detection because text is the most common form of social media. However, the content information in social media is short, noisy, and unstructured with incorrect spellings and symbols, and this impacts the performance of some traditional machine learning methods based on vocabulary knowledge. For this reason, we propose a Char-CNNS (**Char**acter-level **C**onvolutional **N**eural **N**etwork with **S**hortcuts) model to identify whether the text in social media contains cyberbullying. We use characters as the smallest unit of learning, enabling the model to overcome spelling errors and intentional obfuscation in real-world corpora. Shortcuts are utilized to stitch different levels of features to learn more granular bullying signals, and a focal loss function is adopted to overcome the class imbalance problem. We also provide a new Chinese *Weibo* comment dataset specifically for cyberbullying detection, and experiments are performed on both the Chinese Weibo dataset and the English Tweet dataset. The experimental results show that our approach is competitive with state-of-the-art techniques on cyberbullying detection task.

KEYWORDS

convolutional neural networks, cyberbullying detection, social network, text classification

1 | INTRODUCTION

Cyberbullying is bullying that takes place over digital devices such as cell phones, computers, and tablets.[1] Cyberbullying can be achieved in various ways, such as sending a message containing abusive or offensive content to a victim, and some labeled posts are shown in Table 1. In a 2018 statistical report, during the 2015-16 school year, approximately 12% of public schools reported that students had experienced cyberbullying on and off campus at least once a week, and 7% of public schools reported that the school environment was affected by cyberbullying.[2] It can create negative online reputations for victims, which will impact college admissions, employment, and other areas of life, and can result in even more serious and permanent consequences such as self-harm and suicide.[3] Cyberbullying events are hard to recognize. The major problem in cyberbullying detection is the lack of identifiable parameters and clearly quantifiable standards and definitions that can classify posts as bullying.[4] As people spend increasingly more time on social networks, cyberbullying has become a social problem that needs to be solved, and it is very necessary to detect the occurrence of cyberbullying through an automated method.

Our research focuses on textual cyberbullying detection because text is the most common form of social media. In text-based cyberbullying detection, capturing knowledge from text messages is the most critical part, but it is still a challenge. The first challenge that cannot be ignored is dealing with unstructured data. The content information in social media is short, noisy, and unstructured with incorrect spellings and symbols[5] such as the instances in Table 1. Social media users intentionally obfuscate the words or phrases in the sentence to evade manual and automatic detection as in R3. These extra words will expand the size of the vocabulary and influence the performance of the algorithm. Emojis made up of symbols such as :) in R4, which definitely convey emotional features, are always hard to distinguish from noise.

TABLE 1 Some instances in dataset

R1	Sassy.. More like trashy
R2	I HATE KAT SO MUCH
R3	Kat, a massive c*nt
R4	Shut up Nikki … That is all :)

Another key challenge in cyberbullying research is the availability of suitable data, which is necessary for developing models that can classify cyberbullying. There are some datasets have been publicly available for this specific task such as the training set provided in CAW 2.0 Workshop and the Twitter Bullying Traces dataset.[6] We provide a new Chinese *Weibo* comments dataset that was collected from Sina Weibo comments specifically for cyberbullying detection, and we will detail our data collection methods and annotation principles in Section 4. Our dataset will be made available on `Github`.*

Since cyberbullying detection has been fully illustrated as a natural language processing task, various classifiers have been masterly improved to accomplish this task, including the Naive Bayes,[7] the C4.5 decision tree,[8] random forests,[9] SVMs with different kernels, and neural networks classifiers.[6] A variety of feature selection methods have also been carefully designed to improve the classification accuracy.[9-13] However, previous data-based works have relied almost entirely on vocabulary knowledge, and so, the challenges that are posed by unstructured data still exist.

Our work proposes a Char-CNNS (**Char**acter-level **C**onvolutional **N**eural **N**etwork with **S**hortcuts) model to identify whether the text in social media contains cyberbullying. This work proposes a new model with a character-level convolutional neural network to detect cyberbullying. Our model is essentially a classifier based on character-level convolutional neural network (CNN) with varying size filters. We use characters as the smallest unit of learning, enabling the model to learn character-level features to overcome the spelling errors and intentional obfuscation in data. We utilize shortcuts to stitch different levels of features to learn more granular bullying signals, and we adopt a focal loss function to overcome the class imbalance problem in the datasets.[14]

Our contribution We provide a new Chinese *Weibo* comments data set of 19K comments specifically for cyberbullying detection. We also propose a new model to identify whether the text in social media contains cyberbullying. Our experiments are performed on both the Chinese Weibo dataset and the English Tweet dataset, and the results show that our approach is competitive with the state-of-the-art techniques on the cyberbullying detection task.

2 | RELATED WORK

2.1 | Cyberbullying detection

Traditional studies on cyberbullying stand more on a macroscopic view. These studies focused on the statistics of cyberbullying, explored the definitions, properties, and negative impacts of cyberbullying and attempted to establish a cyberbullying measure that would provide a framework for future empirical investigations of cyberbullying.[15-18]

As cyberbullying has captured more attention, various methods have been used for the detection of cyberbullying in a given textual content. An outstanding work is the one by Nahar et al. Their work used the Latent Dirichlet Allocation (LDA) to extract semantic features, TF-IDF values and second-person pronouns as features for training an SVM.[19] Reynolds et al used the labeled data, in conjunction with the machine learning techniques provided by the Weka tool kit, to train a C4.5 decision tree learner and instance-based learner to recognize bullying content.[8] Xu et al showed that the SVM with a linear kernel using unigrams and bigrams as features can achieve a recall of 79% and a precision of 76%.[6] Dadvar et al took into account the various features in hurtful messages, including TF-IDF unigrams, the presence of swear words, frequent POS bigrams, and topic-specific unigrams and bigrams, and the approach was tested using JRip, J48, the SVM, and the naive Bayes.[10] Kontostathis et al analyzed cyberbullying corpora using the bag-of-words model to find the most commonly used terms by cyberbullies and used them to create queries.[20] In the work of Ying et al, the Lexical Semantic Feature (LSF) provided high accuracy for subtle offensive message detection, and it reduced the false positive rate. In addition, the LSF not only examines messages, but it also examines the person who posts the messages and his/her patterns of posting.[12] As the use of deep learning becomes more widespread, some deep learning-based approaches are also being used to detect cyberbullying. The work of Agrawal and Awekar provided several useful insights and indicated that using learning-based models can capture more dispersed features on various platforms and topics.[21] The work of Bu and Cho provided a hybrid deep learning system that used a CNN and an LRCN to detect cyberbullying in SNS comments.[22]

Since previous data-based work relied almost entirely on vocabulary knowledge, the challenge posed by unstructured data still exists. Some works observed that the content information in social media has many incorrect spellings, and in some cases, the users in social media intentionally obfuscate the words or phrases in the sentence to evade the manual and automatic detection.[23,24] These extra words will expand the vocabulary and affect the various performances of the algorithm. Waseem and Hovy performed a grid search over all possible feature set combinations. They found that using character n-grams outperforms when using word n-grams by at least 5 F1-points using similar features,[25] and it is a creative way to reduce the impacts of misspellings. Al-garadi et al used a spelling corrector to amend words, but we believe that some mistakes in this particular task scenario hide the speaker's intentions and correcting the spelling will destroy the features in the original dataset.[26] Zhang et al

* https://github.com/NijiaLu/BullyDataset

innovatively attempted to use phonemes to overcome deliberately ambiguous words in their work. However, some homophones with different meanings will get the same expression after their conversion, and thier method scannot solve some misspellings that have no association in thier pronunciations.[24]

Previous psychological and sociological studies suggested that emotional information can be used to better understand bullying behaviors, and thon emoticons in social text messages conveyed the emotions of users.[27] Dani et al presented a novel learning framework called Sentiment Informed Cyberbullying Detection (SICD), which leveraged sentiment information to detect cyberbullying behaviors in social media.[23] Unfortunately, in the past cyberbullying detection work, almost no work took into account these special symbols. As a common preprocessing technique, removing symbols and numbers destroys the features of the emojis in the original dataset.

We believe that spelling mistakes can be learned. Most of the spelling mistakes have an edit distance of less than 2, and there is a certain regular pattern, which is related to people's pronunciation habits and the key distribution on a keyboard.[28,29] In addition, on social networks, in order to convey a special meaning, some spelling mistakes are customary and common. Almost all factors suggest that these errors that we regarded as noise in previous works can be memorized by learning the combinations of characters. We use characters as the smallest unit since working on only characters has the advantage of being able to naturally learn unusual character combinations such as emoticons.[30]

2.2 | Convolutional neural networks

Convolutional neural network (CNN), originally created for image processing, have performed very well in natural language processing (NLP), especially in sentiment analysis and question classification. Convolutional neural networks with end-to-end training were used in NLP for the first time in other works.[31,32] Their groundbreaking work introduced a new global max-pooling operation, which has been proved to be effective for text, as an alternative to the conventional local max-pooling of the original LeNet architecture.[33] As a brilliant variant, Kim proposed a simpler multichannel architecture with varying size filters.[34] Kalchbrenner et al proposed a wide convolution operation and dynamic k-max pooling structure to handle variable-length input sentences.[35] These classical works have proved that CNNs have excellent performance in text classification tasks, and they are constantly being updated by the efforts of many researchers.

In NLP, capturing features from text is a critical part. Thanks to the work on distributed representations, which are also known as word embeddings, the initialization of embedding vectors has become more efficient with the help of open tools such as `word2vec`[†] and `Glove`.[‡] Essentially after word embedding, vocabularies are combined into a set of vectors in a relatively low-dimensional space, and the distance between these vectors is determined by their semantic relationship. However, some works further noted that word-based input representations may not be very well adapted to social media inputs such as Twitter, where the token usage may be extremely creative.[36]

In addition to word-level inputs, character-level inputs can also build a language model. The idea of character-level language modeling comes from signal processing. The grammar and word semantics in the text are simply ignored because it is widely believed that the model can capture this grammar and word semantic information. The challenge of character-level language modeling is that it requires a large amount of data and enough training time to make the model smart enough to extract the grammatical information and word semantic information from the text. In addition, it also requires data expansion to avoid generalization errors. The work of Zhang et al is the first to apply a CNN only on characters,[30] and their innovative work indicated that deep convolutional neural networks do not rely on word knowledge. They also suggested that ConvNets may have better applicability to real-world scenarios. They coded the characters in the alphabet to quantify the characters and fixed the input length to 1014 since it seems already capture most of the text of interest. However, their work focused on large-scale datasets and the parameters they designed are oversized for a corpus such as the comment text of a social media platform. Johnson and Tong compared the performance of a CNN text classification model on the word-level and character-level. In their results, the shallow word-CNNs generally achieved better error rates and higher speed than those of the very deep char-CNNs, but they used more parameters and therefore require more memory.[37] Especially for corpus of a social media, the method uses a huge embedded table that contains many noise words.

3 | METHOD

In this section, we introduce the design of character-level convolutional network with shortcuts (Char-CNNS) for cyberbullying detection. The model architecture, shown in Figure 1, is a variant of the CNN architecture with shortcuts. Our model is essentially a classifier based on character-level convolutional neural network (CNN) with varying size filter. We skillfully utilize shortcuts to stitch different levels of features to learn more granular bullying signals, and we adopt focal loss function to overcome the class imbalance problem in dataset.[14] It should be noted that for the sake of intuitive effects, our schematic shows a simplified version of the model, considering only a single filter size each layer and ignoring the depth of the vector. Now, we introduce the details of our model in a hierarchical order in Section 3.1.

[†] http://code.google.com/p/word2vec/
[‡] https://nlp.stanford.edu/projects/glove/

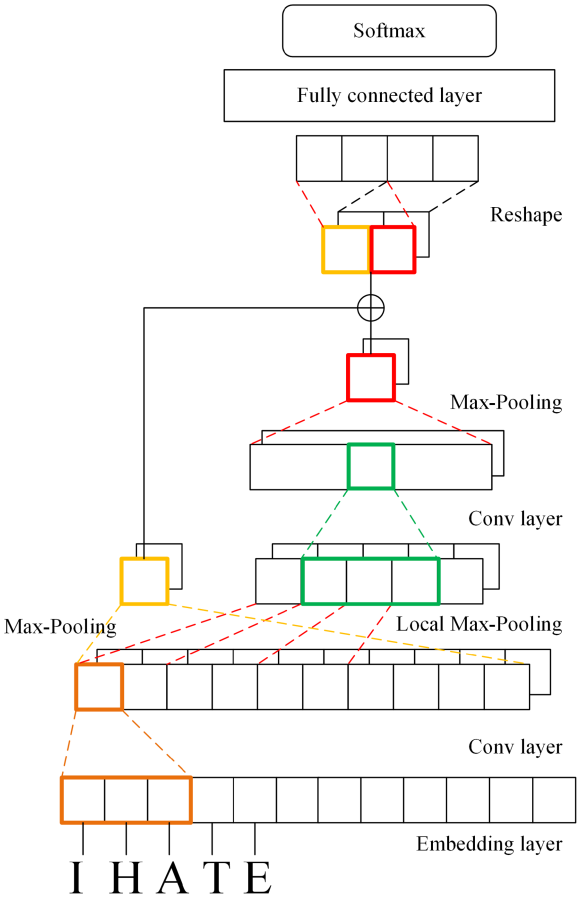

FIGURE 1 The simplified diagram of our Char-CNNS model

3.1 | Model design

In embedding layer, differently from previous alphabet-based schemes, all characters are randomly initialized on normal distribution for the reason of multilingual versatility. Since we use character $x_i \in \mathbf{R}^d$ to be the smallest unit of a token, each message will be seen as the combination of the character as $x_{1:n} = \{x_1; x_2; ...; x_n\}$, where d is the dimension of character. After embedding, the token $x_{1:n}$ will enter the next convolutional layer for feature extraction.

The first convolutional layer captures the feature map composed of the features of neighboring characters. A convolution operation can be described as

$$c_i^{(1)} = f(w \cdot x_{i:i+h-1} + b). \tag{1}$$

Here, $w \in \mathbf{R}^{hxd}$ is a filter, and h is window size of the filter. We use the multiple filters with varying window sizes because it can attention on the features in different lengths, but the specific size setting depends on the corpus used. $b \in \mathbf{R}^d$ is a bias term and f is a nonlinear activate function *Relu*. The superscript $^{(1)}$ indicates that it is the output of the first layer of convolution. After the filter w slides over the token $x_{1:n}$ to complete the convolution operation in the first layer, the feature map $\mathbf{c}^{(1)} = \{c_1^{(1)}, c_2^{(1)}, \ldots, c_{n-h+1}^{(1)}\}$ is produced from each token.

The nonoverlapping *local max-pooling operation*[33] is used to build our multilayer convolution structure. We use the 1-D vision as

$$h_i^{(1)} = max\left\{c_{i\times k-k+1}^{(1)}, \ldots, c_{i\times k}^{(1)}\right\}, \tag{2}$$

where $i = \{1, \ldots, \lfloor \frac{n-h+1}{k} \rfloor\}$ and the k is the size of each chunk. Informally, it first divides the feature map into chunks of fixed size k and then takes the maximum value $h_i^{(1)}$ in each chunk. Therefore, the output of this layer is $\mathbf{h}^{(1)} = \{h_1^{(1)}, h_2^{(1)}, \ldots, h_{\lfloor \frac{n-h+1}{k} \rfloor}^{(1)}\}$. This kind of pooling roughly retains the location information in their pooled features.

The second convolution operation is as same as the first layer except for the filter size. The filter size has been expanded because we expect to learn more context information and get the feature map $\mathbf{c}^{(2)}$. Then, we use global max-pooling to get the most critical features in a post. The *global max-pooling*, which used to get the maximum value, is defined as

$$\mathbf{h}^{(2)} = \max\{\mathbf{c}^{(2)}\}. \tag{3}$$

It has the advantage to naturally deal with variable sentence length.[32]

Shortcut structure is adapted to make use of the feature from different layers. That is, the feature map will concurrently enter into the fully connected layer (after global max-pooling), and the second convolutional layer as shown in Figure 1. Actually, we utilize global max-pooling to get the maximum of feature map $\mathbf{c}^{(1)}$, written as $\mathbf{h}^{(s)} = \max\{\mathbf{c}^{(1)}\}$. The $\mathbf{h}^{(s)}$ will directly *shortcut* concat with the output $\mathbf{h}^{(2)}$ of the second layer and be reshaped as $\{\mathbf{h}^{(s)}, \mathbf{h}^{(2)}\}$ and be sent to the fully-connected layer as input x.

Similar to traditional neural networks, the *fully connected* layer is defined as

$$y = f(\mathbf{W} \cdot x + b), \tag{4}$$

where the W will be randomly initialized on normal distribution. Ultimately, the softmax layer outputs the probability distribution over labels

$$p_i = \frac{\exp(y_i)}{\sum_{k=1}^{n} \exp(y_k)}. \tag{5}$$

3.2 | Loss function and regularization

We use *Focal Loss*[14] that adds a factor $(1 - p_t)^\gamma$ to the standard *Cross Entropy (CE)* criterion. The focal loss is designed to address the one-stage object detection scenario in which there is an extreme imbalance between foreground and background classes during training. However, it can also be used as a technique to deal with category imbalances in natural language processing task. With tunable focusing parameter $\gamma \geq 0$, the focal loss is defined as

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t), \tag{6}$$

where $y \in \{0, 1\}$ is the label and $p \in [0, 1]$ is the models estimated prediction; the factor $-\log(p_t)$ is the rewriten CE by defining p_t

$$p_t = \begin{cases} p, & y = 1 \\ 1 - p, & otherwise. \end{cases} \tag{7}$$

For regularization, we employ dropout on the penultimate layer, which randomly drops out the hidden unit with a proportion p and additionally constrain l_2-norms of the weight vectors with a parameter λ.[38]

4 | EXPERIMENT

4.1 | Dataset description

We provide a new Chinese *Weibo* comment dataset specifically for cyberbullying detection, and we will detail our data collection methods and annotation principles in the following paragraph. To better evaluate our method, we compare it with the previous outstanding approaches and verify its cross-language versatility, and the English Tweet data set that was kindly provided by Waseem and Hovy[25] is also introduced.

4.1.1 | Chinese Weibo dataset

SINA Weibo is a popular social media platform in China, with more than 340 million active users according to their first quarter earnings report. It is a lightweight blog where bloggers can post their updates, and all netizens can participate in comments. The comments of netizens on blogs are our target data.

To obtain more comments containing bullying content, we have selected the weibos that belong to more than 20 celebrities who have bad reputations or who have experienced some vicious incidents. There are large numbers of comments under their Weibo posts, including various opinions such as support, opposition, and even attacks and abuse, some of which are consistent with the definition of cyberbullying. These data are ideal for the experiment on identifying bullying content because public figures have the potential to become cyberbullying victims.

In addition, we have added a supplementary dataset to compensate for the lack of semantic scenarios because the object of the aboves comment is almost always the person. As a supplement, we also collected the comments on some brands, games, and social news that involve a wide range of fields to enrich the objects and scenes in the dataset.

All data are manually annotated by three members who are familiar with Weibo and have a certain understanding of the target bloggers that we crawled. It is unrealistic to annotate all comments according to a quantitative principle, but our group members follow several rules during the process. The criteria partially refer to the work of Waseem and Hovy Hovy[25] and other research on cyberbullying.[15-18]

A Weibo comment is labeled as bullying if it does any of the following:

1. Uses a sexist, racial, or geographical slur;
2. Uses swear words or humiliation to blame someone without a well founded argument;

TABLE 2 The statistics of Weibo dataset

Dataset	Data sources	Comments	Bullying	Distribution
Celebrity	Stars, celebrities, anchors, players	12 141	4000	32.9%
Supplement	Brands, games, and social news	7254	1067	14.7%
Total	-	19 395	5067	26.1%

TABLE 3 The instances of Weibo dataset

Original text	English translation
我感觉骂你都脏了我的嘴，从来没有见过如此厚颜无耻之人	It will dirty my mouth to blame you, I have never seen such a brazen person.
USERNAME 该下十九层地狱	*USERNAME* should go to the next nineteenth hell
这女的就是zuo	This women is just *zuo* (a Chinese phonetic symbol for troublesome or bitchy)
…	…

TABLE 4 The statistics of tweet dataset

Label	Number of tweets
Sexism	3383
Racism	1972
Neither	11 559
Total	16 914

3. Blatantly misrepresents the truth or seeks to distort the views on a minority with unfounded claims;
4. Calls for violence towards or threaren on a minority;
5. Contains attacks on someone's appearance, body, or family members;
6. Is one of repeated negative comments, or it calls on others to join the attack;
7. Imposes a nickname that others are unwilling to accept or is insulting.

It is important to note that there is a difference between this task and sentiment analysis and not all negative comments that marked as bullying followed our criteria. Other offensive content that is not in the criteria will be subjectively determined by the annotators based on the degree of maliciousness.

Since bullying only accounts for a small proportion of the total data, we manually adjusted the distribution of the data by discarding some comments that were not related to bullying. The detailed statistics of the data are shown in Table 2. `Data Source` indicates the object of these comments. The Table also shows the numbers of comments in the dataset, the number of comments marked as bullying, and their percentages. Some Chinese Weibo data instances and their English translations are shown in Table 3.

For preprocessing, we use jieba[§] to segment the Chinese corpus on the word-level. All legal characters under utf-8 encoding are preserved, and all names are replaced by *USERNAME* for privacy purposes. When a comment text is considered to be bullying by more than half of the annotators, we label this text data as 1, and others are labeled as 0. The detection task is a supervised two-category task.

4.1.2 | English tweet dataset

Twitter is a popular online news and social networking service on which users post and interact with messages, which are known as *tweets*. Some researchers have used Twitter's public API[¶] to collect the comment text from the social media and manually label whether the text contains cyberbullying.[13,26,39]

To better evaluate our method, we compare it with the previous outstanding approach and verify its cross-language versatility, and we also use the English data set that was kindly provided by Waseem and Hovy in 2016.[25] The dataset consists of tweets collected over the course of two months, and the statistics of the manually annotated data are described as Table 4 Waseem and Hovy retrieved 136 052 tweets in total and annotated 16 914 tweets; 3383 of tweets were labeled as sexist content, 1972 tweets were labeled as racist content, and 11 559 tweets were labeled as neither sexist nor racist, and their dataset was made available as tweet IDs and labels at Github.[#]

For preprocessing, www.* and https://* was converted to *URL*, and @username was converted to *USRNAME*. In detail, we treated *URL* and *USRNAME* as single characters each because we did not require additional knowledge about the combined features of these two tags introduced by the outside. Different from conventional preprocessing, we kept the capitalization and did not stem the text and reserved special characters and symbols so that our model can learn the extra knowledge from this abnormal information.

Since there are three classes (sexism, racism, and none) in the dataset, we designed experiments to complete the two-class and multiclass tasks. For binary classification, both sexism and racism labels will be equally regarded as bullying. In the multiclassification, we used the original

[§]An excellent Python Chinese word segmentation component. https://github.com/fxsjy/jieba
[¶]https://apps.twitter.com/
[#] http://github.com/zeerakw/hatespeech

TABLE 5 Experiment result

Dataset	Method	Precision	F-measure	Recall
Weibo	TF-IDF + SVM	0.788	0.671	0.584
	Word n-gram + LR[25]	0.783	0.525	0.396
	Char n-gram + LR[25]	**0.795**	0.670	0.579
	Word-CNN[34]	0.706	0.702	0.632
	Char-CNNS	0.790	**0.716**	**0.698**
Tweet	TF-IDF + SVM	0.787	0.656	0.629
	Word n-gram + LR[25]	0.723	0.588	0.545
	Char n-gram + LR[25]	0.728	0.673	0.689
	Word-CNN[34]	0.752	0.725	0.684
	Char-CNNS	**0.810**	**0.742**	**0.705**

labels. In the comparison experiment with the baseline, we did not balance the dataset and use the real dataset as much as possible. However, in the discussion, we will mention the impact of oversampling and the focal loss function on dealing with unbalanced datasets.

4.2 | Hyperparameters and training

The hyper\parameter settings of neural networks may depended on the dataset being used. We used rectified linear units, filter windows h of (3,4,5) and (7,8) on the two layers with 256 feature maps each. We used the chunk of size 3 for local max-pooling, and set the dropout rate (p) of 0.5 and l_2 constraint λ of 3. For the parameters of focal loss function, we followed the settings of Lin et al (α of 0.5 and γ of 2).[14]

The network was trained using labeled data with back-propagation. We applyed Xavier initializer[40] to initialize the **W** in fully connected layer. Training was done over shuffled batches with the Adam update rule.

4.3 | Comparison of methods

We first introduce a few baseline methods. Since we are more concerned with character-level features, we just use the random embedding and have not introduced word embedding steps.

TF-IDF+SVM: The performance of this traditional machine learning algorithm in text categorization tasks is still very impressive. Term frequency–inverse document frequency (TF-IDF) is a typical feature used for text classification, and support Vector Machine (SVM) is a generalized linear classifier for binary classification of data in supervised learning. We use the `scikit-learn` library[‖] provided on Python to implement the TF-IDF+SVM method for Cyberbullying detection tasks.

N-grams+LR: It is the state-of-the-art method,[25] which uses character n-grams for hate speech detection. In order to pick the most suitable features, they perform a grid search over all possible feature set combinations, and use a logistic regression (LR) classifier and 10-fold cross validation to quantify their expressiveness.

Word-CNN: We leverage CNNs for cyberbullying detection use the model proposed by Kim.[34] We have followed their hyperparameter settings and used random embeddings on word-level. Maintaining the same steps, for datasets without a standard dev set, we randomly select 10% of the training data as the dev set and evaluate on it.

Char-CNNS: For binary classification, both sexism and racism labels will be equally regarded as bullying, and for multiclassification, we reserved the original labels. The detail of setting about hyperparameters and training can be found in Section 4.2.

Considering the generalization performance of the method, the test sets is independent on train set, which gives a certain reduction in the final test results. For TF-IDF+SVM and Char n-grams+LR, we adopted 10-fold cross-validation, while we randomly selected 20% of the data as the test set and selected 10% of others as the dev set on neural networks based model.

4.4 | Results and discussion

The compared results of the different methods with respect to P (Precision), F (F-measure), and R (Recall) are shown in Table 5. It shows that our method has certain advantages in different indicators compared with the previous methods. We believe that this advantage comes from the design of some of the modules in our model (embedding on the char-level, adding shortcuts, and choosing a more suitable loss function), and we will elaborate on the effects of these modules in the following discussion.

Table 6 shows some instances of the predictions of the different methods that can reflect the advantages of the Char-CNNS and its ability to some extent. Examples R1 and R2 show that the model has a fairly good ability to learn semantics. For R1, all models misjudged the text as bullying except for the model based on CNN. This is because R1 contains words that are prone to misjudgment such as `Arabs` and `women`. These are high-frequency words that appear in the racism and sexism bullying categories and are likely to be mistaken for bullying in models that do not emphasize semantics. Correspondingly, there is no obvious vocabulary that is directly related to bullying in R2, but from the context of the

[‖] https://scikit-learn.org/

TABLE 6 Instance of test results from different methods

	Text	Manual labeling	TF-IDF +SVM	Char n-gram LR	Word CNN	Char CNNS
R1	How will we Arabs ever get to know freedom when half of our societies, women, are not free?	**None**	Bully	Bully	**None**	**None**
R2	U mean like this? Good luck buying a razor sharp enough to shave your hirsute fanny, armpits and legs.	**Bully**	None	None	**Bully**	**Bully**
R3	Karma is a b i t * h	**Bully**	None	**Bully**	None	**Bully**

semantics, there is no doubt that this is a tweet containing bullying. Only the char-level model got the right results on R3, and this is not too surprising since our model is designed for this type of bullying signal recognition.

4.4.1 | The performance of the focal loss function with varying degrees of imbalance

When the numbers of the positive and negative samples are very different, the α parameter in the focal loss function can be adjusted to control the weight of the positive examples, and the parameter γ can be used to adjust the difficulty level of learning the samples. This allows the model to overcome the data imbalance from the perspective of cost functions and to fine-tune the task.

To verify the performance of our method on the unbalanced data set, we hierarchically sampled 10 000 weibos and constructed five datasets with the ratio of positive to negative cases set as 1:1, 1:2, 1:5, 1:10, and 1:20. The F-measure curves of different methods with the different data distributions are shown in the Figure 2. The comparison methods include TF-IDF + SVM (SVM), Char n-gram + LR (LR), Word-CNN with the cross entropy loss function (CNN), and our Char-CNNS model with the focal loss function (Char-CNNS). We also experimented with the focal loss function under different parameter settings, using standard cross entropy loss function (CE) as the baseline.

Compared with other schemes, the Char-CNNS performs relatively well for the different data distributions with respect to the F-measure. When the ratio of positive and negative samples of the data set reaches 1:20, the Char-CNNS has an excellent recall rate compared with other methods, which is 12.4, 13.1, and 5 percentage points higher than that of the SVM, LR, and CNN methods, respectively. FL with parameters can better adapt to multiple distributions than the CE, and the parameter settings of ($\alpha = 0.25, \gamma = 1$) result in better performance on the Weibo corpus. Furthermore, this advantage is more pronounced as the data imbalance increases.

4.4.2 | The influence of char-level

Using the character level can achieve better accuracy in the special task of Cyberbullying detection, and our experimental results confirm this point. Whether the model uses a neural network structure, the character-level performed better than the word-level in this task. The work of Waseem and Hovy found that using character n-grams outperforms using word n-grams by at least 5 F1 percentage points when using similar features.[25] As the baseline for neural network structure, we leveraged the CNN for cyberbullying detection task using the model that was proposed by Kim[34] and followed their hyperparameter settings. Four sets of experiments that involved two inputs and two model frameworks were used to validate our views: (1) word-level + CNN, (2) char-level + CNN, (3) word-level + CNNS, and (4) char-level + CNNS. We used random embedding on char-level and word-level.

In order to clearly describe the detail about training, the accuracy on the dev set is shown in Figure 3. The line charts show that the accuracy of the char-level performance is improved in both the CNN and our model compared to the word-level by at least 1 and 1.6 percentage points for

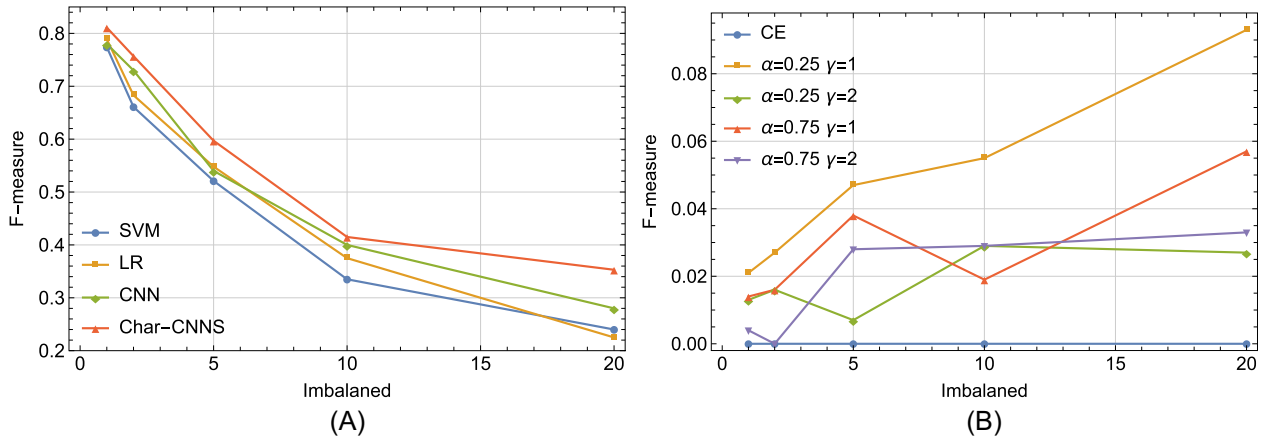

FIGURE 2 The abscissa indicates the ratio of negative and positive samples in the Weibo dataset, reflecting the degree of imbalance in the dataset. Subgraph (A) shows the change in F-measures of different baseline methods when the degree of imbalance in the dataset is various, and subgraph (B) demonstrates the gap of F-measures between cross entropy (CE) and focal foss (FL), reflecting the relative value of the F

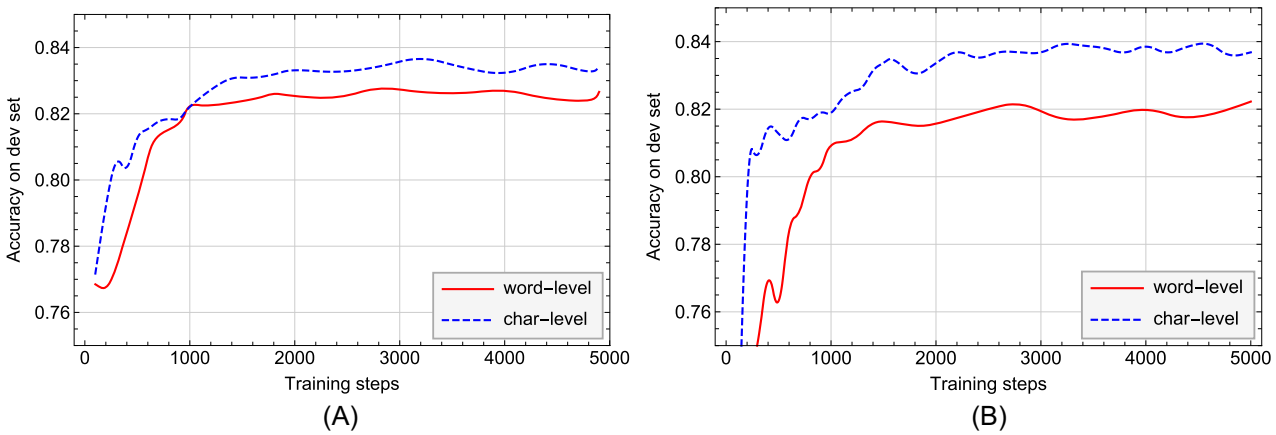

FIGURE 3 Compared the accuracy on the dev set of char-level and word-level features on tweet corpus using CNN (A) and our Char-CNNS (B). For both methods, char-level have better percentage points than word-level

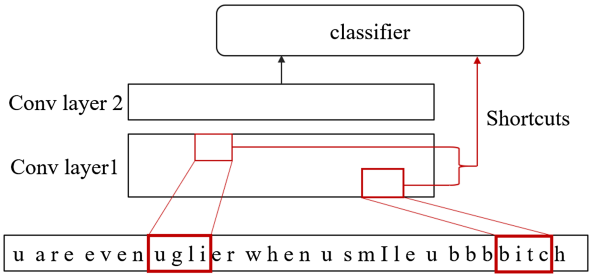

FIGURE 4 Shortcuts use the lowest level of character combination features directly, while CNN is responsible for semantic features

each approach, respectively. In addition, the improvement in the other indicators is also obvious. For the CNN model, the precision on test set increased by 6.0 percentage points, and for Char-CNNS, the P, F1, and R get improved 8.6, 4.0, 1.8 percentage points, respectively, at average by replacing the word-level with char-level. It confirms our point of view that the features on the char-level for the corpus in real-world scenarios perform better than word-level. Working on only characters has the advantage of being able to learn unusual character combinations, and it is a desirable approach to overcome spelling errors, symbols and intentional obfuscation in data. Instance R3 in Table 6 also supports this view.

4.4.3 | How the shortcuts works

We use the shortcuts to stitch different levels of features to learn hybrid bullying signals, and in this paragraph we try to explore its role. In ResNet, the residual structure that is composed of shortcuts is used to deepen the learning of neural network, but we use it to combine the features from different levels. In other words, we use a multilayer neural network to learn the semantic features based on the combination of characters, and the lexical features of these characters themselves will be used in parallel through the Shortcuts. Figure 4 simply shows the construction of the shortcuts, which makes some of the more intuitive character-level features directly available.

A simple contrast experiment is designed to verify the effect of shortcuts. The baseline method used a basic two-layer CNN to capture features after two layers of convolution and aggregation as a baseline (we informally call it single features), while our model with shortcuts captured the hybrid features, and we still compared their performance on accuracy on the dev set. The parameters of these two models are kept consistent, and the only difference is whether to use the shortcuts to concat the features obtained from the two layers. The accuracy on the dev set is shown in Figure 5. We found that after adding shortcuts, the accuracy on the dev set did improve about 1.5 percentage point. Moreover, on the test set, the P, F, and R get improved 3.7, 5.0, and 6.8 percentage points, respectively, with the shortcut to construct the hybrid features.

The following reasons may cause the shortcuts to improve the results of the task. While we use characters as the smallest units, increasing the depth of a neural network simply blurs the underlying character composition features from the lower layers, but paradoxically, the character-level does require multiple layers to capture contextual features. However, this problem has been overcome by using features from different layers in parallel. At this point, the parallel consideration of low-level and high-level features can be a compromise. Therefore, we believe that learning hybrid features from multiple layers is effective, especially for the character level. It ensures that the model can learn the semantics and also preserves the surface features composed of the character combinations.

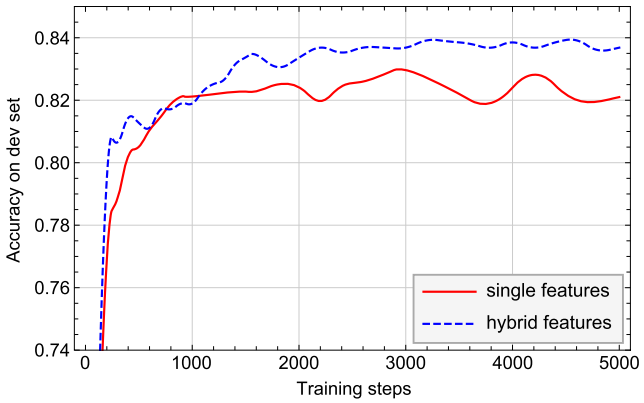

FIGURE 5 The baseline method uses a basic two-layer CNN to capture single features, while our model with shortcuts captures the hybrid features. The parameters of these to models are kept consistent, and the only difference is whether to add the shortcuts to concat the features obtained from the two layers. The experiment results show that adding shortcuts to capture hybrid features will increase the performance of the model on different indicators

5 | CONCLUSION AND FUTURE WORK

We provide a new Chinese *Weibo* comments dataset of 19K comments specifically for cyberbullying detection. All the samples belonging to more than 20 celebrities who have bad reputations or who have experienced some vicious incidents have been selected and manually annotated. These data are ideal for experiments on identifying bullying content, because these public figures have the potential to become cyberbullying victims.

We also propose an automatic solution to identify whether the text in social media contains cyberbullying. It learns the char-level features to overcome spelling errors and intentional obfuscation in data. Shortcuts are utilized to stitch different levels of features to learn hybrid bullying signals, and we adopt a focal loss function to overcome the class imbalance problem in the dataset. These well-designed modules are simple but truly effective, and our approach has better performance with a P, F, and R of 79.0, 71.6, and 69.8, respectively, on the Weibo dataset and 81.0, 74.2, and 70.5, respectively, on the Tweet dataset. Compared with other schemes, the Char-CNNS with the focal loss performs relatively well for different data distributions. The experimental results demonstrates that the character-level embedding and hybrid features from multiple layers increase the performance of cyberbullying detection on social media text.

The experiment shows that a shallow neural network model already works excellently and it has achieved satisfactory results. In the future, we hope to expand its number of layers to explore if there is any further improvement. It is also a worthwhile attempt to adjust the weights of the shortcut branches. In addtion, identifying more types of bullying is also a direction that we want to explore in the future.

ACKNOWLEDGMENTS

This work was partially supported by Zhejiang Province Natural Science Foundation (No. LY18F020017), National Natural Science Foundation of China (No. 61872120).

ORCID

Nijia Lu https://orcid.org/0000-0001-9130-6977
Kim-Kwang Raymond Choo https://orcid.org/0000-0001-9208-5336

REFERENCES

1. StopBullying.gov. https://www.stopbullying.gov/
2. Musu-Gillette L, Zhang A, Wang K, et al. Indicators of school crime and safety: 2017. National Center for Education Statistics and the Bureau of Justice Statistics. 2018.
3. Hinduja S, Patchin JW. Bullying, cyberbullying, and suicide. *Arch Suicide Res*. 2010;14(3):206-221.
4. Sugandhi R, Pande A, Chawla S, Agrawal A, Bhagat H. Methods for detection of cyberbullying: A survey. Paper presented at: 15th International Conference on Intelligent Systems Design and Applications; 2015; Marrakech, Morocco.
5. Baldwin T, Cook P, Lui M, MacKinlay A, Wang L. How noisy social media text, how different social media sources. Paper presented at: 6th International Joint Conference on Natural Language Processing; 2013; Nagoya, Japan.
6. Xu JM, Jun KS, Zhu X, Bellmore A. Learning from bullying traces in social media. Paper presented at: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2012; Montreal, Canada.
7. Freeman DM. Using naive Bayes to detect spammy names in social networks. Paper presented at: ACM Workshop on Artificial Intelligence and Security; 2013; Berlin, Germany.
8. Reynolds K, Kontostathis A, Edwards L. Using machine learning to detect cyberbullying. Paper presented at: 10th International Conference on Machine learning and Applications and Workshops; 2011; Honolulu, HI.
9. Kasture AS. A predictive model to detect online cyberbullying [master's thesis]. Auckland, New Zealand: Auckland University of Technology; 2015.
10. Dadvar M, Ordelman R, de Jong F, Trieschnigg D. Improved cyberbullying detection using gender information. Paper presented at: 12th Dutchbelgian Information Retrieval Workshop; 2012; Ghent, Belgium.

11. Dinakar K, Reichart R, Lieberman H. Modeling the detection of textual cyberbullying. Paper presented at: 5th International AAAI Conference on Weblogs and Social Media; 2011; Barcelona, Spain.
12. Ying C, Zhou Y, Zhu S, Xu H. Detecting offensive language in social media to protect adolescent online safety. Paper presented at: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing; 2012; Amsterdam, Netherlands.
13. Zhao R, Mao K. Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder. *IEEE Trans Affect Comput*. 2017;8(3):328-339.
14. Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. *IEEE Trans Pattern Anal Mach Intell*. 2017;99:2999-3007.
15. Patchin JW, Hinduja S. Bullies move beyond the schoolyard a preliminary look at cyberbullying. *Youth Violence Juvenile Justice*. 2006;4(2):148-169.
16. Robert S, Smith PK. Cyberbullying: another main type of bullying? *Scand J Psychol*. 2008;49(2):147-154.
17. Smith PK, Jess M, Manuel C, Sonja F, Shanette R, Neil T. Cyberbullying: its nature and impact in secondary school pupils. *J Child Psychol Psychiatry*. 2008;49(4):376-385.
18. Tokunaga RS. Following you home from school: a critical review and synthesis of research on cyberbullying victimization. *Comput Hum Behav*. 2010;26(3):277-287.
19. Nahar V, Xue L, Pang C. An effective approach for cyberbullying detection. *Commun Inf Sci Manag Eng*. 2013;3(5):238-247.
20. Kontostathis A, Reynolds K, Garron A, Edwards L. Detecting cyberbullying: Query terms and techniques. Paper presented at: 5th Annual ACM Web Science Conference; 2013; Paris, France.
21. Agrawal S, Awekar A. Deep learning for detecting cyberbullying across multiple social media platforms. Paper presented at: 40th European Conference on IR Research; 2018; Grenoble, France.
22. Bu SJ, Cho S. A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments. Paper presented at: International Conference on Hybrid Artificial Intelligence Systems; 2018; Oviedo, Spain.
23. Dani H, Li J, Liu H. Sentiment informed cyberbullying detection in social media. Paper presented at: Joint European Conference on Machine Learning and Knowledge Discovery in Databases; 2017; Skopje, Macedonia.
24. Zhang X, Tong J, Vishwamitra N, et al. Cyberbullying detection with a pronunciation based convolutional neural network. Paper presented at: 15th IEEE International Conference on Machine Learning and Applications; 2016; Anaheim, CA.
25. Waseem Z, Hovy D. Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. Paper presented at: North American Chapter of the ACL Student Research Workshop; 2016; San Diego, CA.
26. Al-garadi MA, Varathan D, Ravana SD. Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. *Comput Hum Behav*. 2016;63:433-443.
27. Kokkinos CM. The relationship between bullying, victimization, trait emotional intelligence, self-efficacy and empathy among preadolescents. *Soc Psychol Educ*. 2012;15(1):41-58.
28. Damerau FJ. A technique for computer detection and correction of spelling errors. *Commun ACM*. 1964;7(3):171-176.
29. Kemighan MD, Church KW, Gale WA. A spelling correction program based on a noisy channel model. Paper presented at: 13th International Conference on Computational Linguistics; 1990; Helsinki, Finland.
30. Zhang X, Zhao J, Lecun Y. Character-level convolutional networks for text classification. Paper presented at: International Conference on Neural Information Processing Systems; 2015; Montreal, Canada.
31. Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. Paper presented at: 25th International Conference on Machine Learning; 2008; Helsinki, Finland.
32. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. *J Mach LearnRes*. 2011;12(1):2493-2537.
33. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278-2324.
34. Kim Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*. 2014.
35. Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*. 2014.
36. Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks. Paper presented at: 38th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2015; Santiago, Chile.
37. Johnson R, Tong Z. Convolutional neural networks for text categorization: Shallow word-level vs. deep character-level. *arXiv preprint arXiv:1609.00718*. 2016.
38. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. *Comput Sci*. 2012;3(4):212-223.
39. Raisi E, Huang B. Weakly supervised cyberbullying detection with participant-vocabulary consistency. *Soc Netw Anal Min*; 8(1).
40. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Paper presented at: 13th International Conference on Artificial Intelligence and Statistics; 2010; Sardinia, Italy.

How to cite this article: Lu N, Wu G, Zhang Z, Zheng Y, Ren Y, Choo K-K R. Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts. *Concurrency Computat Pract Exper*. 2020;e5627. https://doi.org/10.1002/cpe.5627