Categorizing Offensive Language in Social Networks: A Chinese Corpus, Systems and an Explanation Tool

Xiangru Tang, Xianjun Shen*, Yujie Wang, Yujuan Yang
School of Computer, Central China Normal University, China
National Language Resources Monitoring & Research Center for Network Media, China
Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, China
xjshen@mail.ccnu.edu

Abstract

Recently, more and more data have been generated in the online world, filled with offensive language such as threats, swear words or straightforward insults. It is disgraceful for a progressive society, and then the question arises on how language resources and technologies can cope with this challenge. However, previous work only analyzes the problem as a whole but fails to detect particular types of offensive content in a more fine-grained way, mainly because of the lack of annotated data. In this work, we present a densely annotated data-set COLA (**C**ategorizing **O**ffensive **LA**nguage), consists of fine-grained insulting language, antisocial language and illegal language. We study different strategies for automatically identifying offensive language on COLA data. Further, we design a capsule system with hierarchical attention to aggregate and fully utilize information, which obtains a state-of-the-art result. Results from experiments prove that our hierarchical attention capsule network (HACN) performs significantly better than existing methods in offensive classification with the precision of 94.37% and recall of 95.28%. We also explain what our model has learned with an explanation tool called Integrated Gradients. Meanwhile, our system's processing speed can handle each sentence in 10msec, suggesting the potential for efficient deployment in real situations.

CCL2020

1 Introduction

In modern society, the occupation of offensive language on the online world, such as social media, is becoming a paramount concern. Offensive language differs considerably, ranging from pure abuse to more rigorous types of writing. Thus, offensive language is hard to be automatically identified. However, it's essential to track this; for example, the appearance of offensive language on social media is related to hate crimes in a real social situation. [Müller and Schwarz2018]. Moreover, it can be pretty troublesome to distinguish fine-grained offensive language because few general definitions exist [Davidson et al.2017].

Recently, researchers have proposed some guidelines to identify the type and the attributes of offensive language [Zampieri et al.2019a]. However, the online world's offensive language is a general category containing specific examples of profanity or insult. In our work, "Offensive language" in the online world is defined in more detail and fine-grained. And to the best of our knowledge, though offensive language identification being a burgeoning field, there is no data-set yet for Chinese.

"Offensive" is pretty much something people identify as against morals, very inappropriate, or disrespectful. However, "offensive" is a broad general term and does not define the precise extent or the limits of its application. Thus, we classify the term "offensive" into three categories: "insulting," "antisocial" and "illegal" through stepwise refinement. "Insulting" is something rude, insensitive and/or offensive, directed at another person or group of people. This emphasizes that the content is a direct attack against specific others. "Antisocial" is harmful to organized society, or the language describes a behavior deviating from the social norm for long. "Illegal" language means it violates the language policy. Where the

* Corresponding author.

©2020 China National Conference on Computational Linguistics
Published under Creative Commons Attribution 4.0 International License

Proceedings of the 19th China National Conference on Computational Linguistics, pages 1045-1056,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

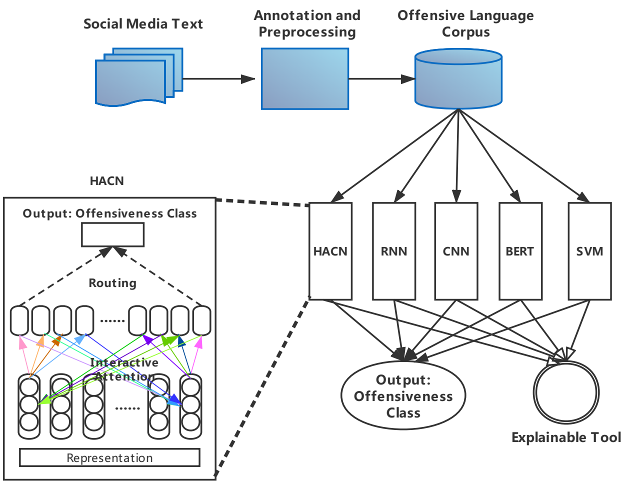

Social Media Text
Annotation and Preprocessing
Offensive Language Corpus
HACN
Output: Offensiveness Class
Routing
Interactive Attention
Representation
HACN
RNN
CNN
BERT
SVM
Output: Offensiveness Class
Explainable Tool

Figure 1: Data Processing Pipeline System and The architecture of Hierarchical Attention Capsule Network (HACN).

language policy refers to the government through legislation or policies to formally decide how languages are used. However, the language policies of each country are not completely consistent.

Thus, two questions arise a) how LRs can cope with the large numbers of offensive language in the online world, and b) can LT provide means to process and respond promptly to such language data streamed in a huge amount at high speed? Firstly, there is no existing data-set for the Chinese language to provide for correctives of hate speeches, cyberbullying, or fake news. Then, current methods can not produce highly precise results for detecting offensive content and behaviors. Also, They used inflexible proprietary APIs, which is hard to reproduce. On the other hand, there is a real need for methods to detect and deal with online words quickly because of the enormous amount of data created every day.

In this context, we present a sizeable Chinese classification corpus of offensive language called COLA. Then, we employ a deep dilated capsule network to extract hierarchical structure. We further design hierarchical attention to aggregate and fully utilize information within a hierarchical representation. Correctly, each sentence is embedded into capsules and incrementally distilled into task-relevant categories during the hierarchical attention process. What is more, we present an explanation tool, which proves that our work for the Chinese language seizes the pattern of offensive language in some points, and almost correctly identifies different varieties of offensive language, like hate speech and cyberbullying.

In summary, our work aims at answering the two questions a) how LRs can cope with a huge amount of offensive language in the online world, and b) can LT provide means to process such language data at high speed? The major contributions are highlighted as follows:

- We describe COLA, the first Chinese offensive language classification dataset. COLA is designed to study how language resources and technologies can cope with this offensive language challenge in the online world. It is now publicly available.

- We propose a hierarchical attention capsule network(HACN), where the hierarchical attention mechanism is introduced to model the hierarchical structure. It is inspired by capsule, with modifications to handle the words explicitly.

- We show that our HACN model surpasses state-of-the-art methods for classification on COLA. Furthermore, our presented explanation tool clearly explains what our model has learned.

2 Related work

2.1 Corpus

Some previous works have discussed how to identify the offensive language, but in that literature, the offensive language is ranging from aggression to cyberbullying, toxic comments, and hate speech. In the

CCL2020

following, we explain each of these open public challenges briefly.

SemEval-2019 Task: In Task 6 of SemEval 2019, they propose three separate sub-tasks. A subtask is Offensive Language Selection, the other is Categorization of Offensive Language, and the last is Offensive Language Target Recognition. SemEval-2019 Task 6 is called OffensEval, and the collection methods of their data are explained in [Zampieri et al.2019b]. Additionally, it collected more than 14100 posts of sentences.

Aggression identification (TRAC): The TRAC study [Kumar et al.2018] provided players with a data set containing a training set and a validation set.They are composed of 15,000 Facebook posts and comments annotated in English and Hindi. For the test set, two different sets are used, one from Facebook and the other from Twitter. It aims at distinguishing three types of data: non-aggressive, covert aggressive, and over-aggressive.

Hate speech recognition: In [Kwok and Wang2013, Burnap and Williams2015, Djuric et al.2015], they present a Abusive language selection task. Specially, [Davidson et al.2017] provided the hate speech recognition data set, which contains more than 24000 English tweets marked as non-offensive, hate speech, and profanity.

Offensive language: The data-set provided by GermEval [Wiegand et al.2018] focused on offensive language recognition in German tweets. The study showed a data set of more than 8,500 tagged tweets. This data set is used to perform binary classification task of distinguishing between offensive and non-offensive information. Besides,the second task divided offending tweets into three categories: profanity, insult, and abuse. While similar to our work, there are three important differences: (i) we have a third level in our hierarchy, (ii) we use different labels in the second level, and (iii) we focus on Chinese.

Toxic comments (Kaggle): Kaggle holds a Toxic Comment Classification Challenge as an open dataset. The dataset in this competition was extracted from the comments of Wikipedia, and it was formed in six categories: toxicant, severe toxic,identity hate, threat, insult,obscene . Moreover, the data set is also employed outside of the competition [Georgakopoulos et al.2018], treated as an external training resource for the TRAC, as mentioned above [Fortuna et al.2018].

However, each of these tasks tackles a particular challenge of detecting offensive language. Thus, we present a new dataset, hoping it could become a valuable resource for improving offensive language categorizing.

CCL2020

2.2 Classification

Traditional classification methods are designed with rich features and syntactic structures to achieve the classification task [Jiang et al.2011]. But, these feature-based methods are labor-intensive, and the performance depends largely on the quality of the features. Recently, deep learning methods are becoming popular for aspect-level sentiment classification. Recurrent Neural Networks (RNNs) are the most commonly used technique for this task [Tang et al.2015]. The attention mechanism is further introduced to model the target-context association [Wang et al.2016]. Recently, CNN-based models have shown the strengths inefficiency to tackle the aspect-level sentiment classification [Xue and Li2018, Huang and Carley2019, Li et al.2018]. However, all the previous methods utilize static pooling operation or attention mechanism to locate the keywords, which fails to handle the overlapped features. We introduce vector-based feature representation and feature clustering to address this.

Capsule network was proposed to improve the representational limitations of CNN and RNN by extracting features in the form of vectors. The technique was firstly proposed in [Hinton et al.2011]. But is mainly devised for the image processing domain. Introducing capsules allows us to utilize a routing mechanism instead of pooling operation to generate high-level features, which is a more efficient way for features encoding. The routing module is able to cluster features in an iterative way, which achieved impressive performance recognizing highly overlapped digits. Several types of capsule networks have been proposed for natural language processing. [Zhao et al.2018] investigated capsule networks for text classification. They also found that capsule networks exhibit significant improvement when transferring single-label to multi-label text classification. However, interactive word-level attention is not considered in these typical capsule routing methods.

3 Data Collection

In this section, we describe the data set and how we annotate data set.

3.1 Overview

Data	Train	Test	Valid	Total
Neutral	5357	1700	1546	8603
Insulting	5075	1660	1493	8228
Antisocia	841	303	218	1362
Illegal	327	96	91	514
Total	11600	3759	3348	18707

Table 1: Statistics of the four classes in COLA data. Number of sentences in train set, test set, valid set.

We create a large-scale data-set that annotates offensive texts in Chinese. The texts are crawled from Youtube and Weibo: 18.7k comments in total. Three annotators categorised these texts in four classes: neutral, insulting, antisocial, and illegal. We build a Chinese dataset from social media that people can communicate on Internet, such as Sina Weibo[0] and YouTube comments. Our released COLA contains user-generated comments from different social media platforms, and as we know, it is the first of its kind. And, the dataset is marked as capture different types of offensive language. We propose four automatic classification systems, each designed to work for the Chinese language.

CCL2020

3.2 Data Acquisition

With more than 1000 comments and more than 10000 views as the thresholds, we selected 20 popular Chinese videos from YouTube. Furthermore, from the comments below the video are crawled through Google YouTube V3 API, which is offered by Google for researchers to collect comments. And a total of 20000 comments were received. We store the 20000 comments, and then we clean the data. We first convert the traditional Chinese character in the data set to simplified Chinese characters, and then filter out the useless data with messy codes and HTML tags.

There are some technologies we employ to crawl the data. Firstly, we retrieve 81718 Chinese sentences from Weibo and YouTube reviews in JSON format, and contain information such as timestamp, URL, text, user, re-tweets, replies, full name, id, and likes. Extensive processing is carried out to remove all the noisy sentences. We apply the following pre-processing steps: the documents are tokenized using NLTK, the URLs and mentioned users are removed, and all letters are converted to lower-case. As a result, a dataset of 18,707 offensive language sentences is created. Nevertheless, social media companies all have some methods to prevent crawlers. These methods can be divided into three categories: analyzing the headers of web page requests, monitoring the behavior of users visiting the website, and adjusting the directory and data loading methods. Corresponding to that, we adopt three approaches to crawl the data. For the first one, we could directly add HEADERS and REFERER to the code to bypass the check. And the same IP visits the same page multiple times in a short period, or the same account performs the same operation multiple times in a short time may cause the second one situation. For this situation, we can use the IP proxy to resolve it. We can use a browser to analyze the requests for the last situation. If we can obtain the AJAX request, then we can use the above two methods to resolve and obtain the corresponding data. However, if we cannot get AJAX requests, we can call the selenium + phantomjs framework and call its browser kernel to simulate human operations and JS scripts that trigger the page.

3.3 Annotation

We construct the data-set which comes from the hot issues of YouTube comments and Weibo. And web crawler gets our the data we needed. The data is annotated by three volunteers. After analyzing all the data, more than 18,707 sentences are selected. Then, we remove invalid tokens in the text, like HTML

[0] https://en.wikipedia.org/wiki/Sina_Weibo

tags and emoticons, and treat the text as the preliminary data for hand-operated annotation. After that, the vocabulary was divided into three categories: insulting language, antisocial language, illegal language. Due to the special combination of sensitive words, the standard of language structure is pretty vague. We note that different people may have different understanding of the text during the process of annotation. It means the boundary of the same word may be different. Thus, three people are asked to annotate the same text to ensure accuracy. We should note that inter-annotator agreement and intra-annotator agreement have been considered for the coherence of annotations While annotating.

4 Proposed Methods

In this part, we describe the categorizing task to be performed, how we perform the task and the excellent methods are proposed for especially this task.

4.1 Task

The recognition and classification of offensive language in the online world can be realized as a multiple classification task. In this section, we describe several proposed neural networks in details. The aim of aspect-level sentiment classification is to predict the class y of a sentence. In our task, the $y \in \{Neutral, Insulting, Antisocial, Illegal\}$.

4.2 Baseline Systems

Several baseline models are evaluated in Table 2.

SVM: For training our SVM classifier, scikit-learn[1] machine learning in Python library is used for benchmarking. During our experiments, we carry out 10-fold cross validation. We select the Linear SVM formulation, known as C-SVC and the value of the C parameter is 1.0.

RNN: RNN is the high-efficiency method to solve classification problem in NLP tasks. In this paper, we adopt GRU, which has great superiority compared to LSTM and basic RNN. In the final Multi Layer Perceptron layer, 128 neurons are used for classification. And Sigmoid activation function is applied to the final layer.

CNN: We adopt word-level CNN model which has 1D convolution layer with 150 filters and kernel size 6, dropout 0.2, cross entropy loss funtion and four dense layers with ReLU, tanh, sigmoid and softmax activation respectively.

BERT: BERT has displayed its great advantage of text representation in many NLP tasks. We fine-tune the task-specific components, such as a softmax classifier with BERT or deem BERT model as a feature extractor. First of all, we pack the input features as $H_0 = e_1, \cdots, e_T$, where $e_t (t \in [1, T])$ is the group of the token embedding, position embedding and segment embedding corresponding to the input token x_t. Then the L transformer layers is introduced to refine the characteristics of the token layer by layer. Specifically, the representations $H^1 = h_1^l, \cdots, h_T^l$ at the l-th ($l \in [1, L]$) layer are calculated below:

$$H^l = \boldsymbol{Transformerl}(H^{l-1}) \tag{1}$$

We treat H^L as the contextualized representations of the input tokens.And use them to execute the downstream task's predictions.

4.3 Challenge

However, there are still several limitations of the current approaches for offensive language categorizing. Firstly, little attention has been paid to the imbalance of different classes, which are essential and challenging because in categorizing tasks, it will be hard to capture the critical pattern of a specific class without sufficient data. Since the COLA data-set is unbalanced, the neural network may not have enough training examples of "illegal language" to learn. Consequently, it cannot catch the feature and structure of the "illegal language".

[1] https://scikit-learn.org/stable/

CCL2020

Second, existing research on offensive language detection cannot accurately detect offensive content because one sentence expresses multiple polarities, resulting in overlapped feature representation. The highly over-lapped features will confuse the classifier seriously, and the three types of specific offensive language do not have quite a considerable distinction. However, most existing methods only keep the most potent feature by max-pooling operation or utilize attention mechanisms to find the keywords, which fails to distinguish the over-lapped features.

Third, the dissemination and use of online platforms have grown significantly in every minute. Thus, the speed of the system is what we aspire to enhance, especially in this task. The systems are required to detect quickly and deal with this type of content in a short time. In this way, we should consider both accuracy and speed as the evaluation metric in a real application.

4.4 Hierarchical Attention Capsule Network

An original capsule network is a group of neurons obtained from the output of the convolutional operation performed on word representation h_n^a and h_n^c. So, the output of the capsule is a vector representing different properties of the same objective. The routing method [Hinton et al.2011] is employed in our model, and except for the high-dimensional output M, there is one more activation probability in our capsule.

We have already decided on the outputs of all the capsules C_L in the first capsule layer, and we now want to decide which capsules C_{L+1} to active in the layer above and how to assign each active low-level capsule to one active higher-level capsule. The vector-based features get clustered in the high-level capsules where the outputs of high-level capsules play the role of Gaussians, and the output vectors of low-level capsules play the role of the data points. To establish a semantic relationship model between aspect terms and context, we further devise an interactive attention-based routing mechanism.

Firstly, every primary capsule i is transformed by W_{ij} to cast a vote $V_{ij} = M_i W_{ij}$ for the output of high-level capsule j. Moreover, we can get the mean μ_j of the votes from the input capsules, and the variance σ_j about that mean for each dimension h:

$$\mu_j^h = \frac{\sum_i R_{ij} V_{ij}}{\sum_i R_{ij}} \tag{2}$$

$$\sigma_j^{h2} = \frac{\sum_i R_{ij}(V_{ij}^h - \mu_j^h)^2}{\sum_i R_{ij}} \tag{3}$$

Then, we can calculate the activation probability of capsule j by:

$$c_j^h = \Big(\beta_u + log(\sigma_j^h)\Big) \sum_i R_{ij} \tag{4}$$

$$a_j = \boldsymbol{sigmoid}(\lambda(\beta_\alpha - \sum_h c_j^h)) \tag{5}$$

In there, μ_j^h is the $h^t h$ component of the capsule $j$âs vectorized output M_j, and β_uã β_α are trainable parameters.

Then, for the part of capsule routing procedure, we propose a hierarchical attention to capture the hierarchical representation. In particular, the scaled dot-product attention is used to map a set of key-value pairs and the query to a weight on the word-level token. The queries are the averaged representation of the word representation h_c are transformed to dimension d_k by trainable parameters:

$$\alpha_n = \frac{\boldsymbol{exp}^{\frac{k_n^c \times q_a}{\sqrt{d_k}}}(q_a, k_n^c)}{\sum_{n=1}^{N} \boldsymbol{exp}^{\frac{k_n^c \times q_a}{\sqrt{d_k}}}(q_a, k_n^c)} \tag{6}$$

CCL2020

We use a spread margin loss, L_k for each top-level capsule k to directly maximize the gap between the activation of the target class. Overall, our loss function L is the sum of the losses of all mentioned capsules:

$$L = \sum_{k \neq t} (\boldsymbol{max}(0, m - (a_t - a_k))) \tag{7}$$

5 Explanation Tool

The explanation for an algorithm is importance in some fields, such as clinical and financial decisions, because its results affect people directly. The European Union brings the *right to explanation* regulation [Goodman and Flaxman2017] into force in 2018, which is a right to be given an explanation for an output of the algorithm. For example, a person who applies for a loan and is denied may ask for an explanation. However, machine learning algorithm is data driven, even the algorithm designers have no idea how it works, especially for the deep neural networks with thousands of parameters. Nowadays, a large amount of explanation methods [Guidotti et al.2018, Olah et al.2017, Ancona et al.2017, Koh and Liang2017, Ribeiro et al.2016] have been proposed to reveal the behavior of deep neural networks. Post hoc methods especially the attribution methods, is a big branch of explanation methods, which assign the output score to the contributions of input features.

$$\begin{aligned} \mathrm{IG}(x; F)_i = & \frac{x_i - x'_i}{m} \\ & \times \sum_{k=1}^{m} \frac{\partial F(x' + k/m \times (x_i - x'_i))}{\partial x_i}. \end{aligned} \tag{8}$$

We utilize *iNNvestigate* [Alber et al.2018] a toolbox for explanation methods to evaluate on different tasks. Especially, we employ Integrated Gradients [Sundararajan et al.2017] as a tool, which is similarly to GradInput and computes the average value while the input varies along a linear path from a baseline x' to x. It solve the problem of Sensitivity property violation. The baseline x' is defined by the user and often chosen to be zero. m is the number of steps in the Riemman approximation of the integral. Our explanation tool is released for other researchers to use.

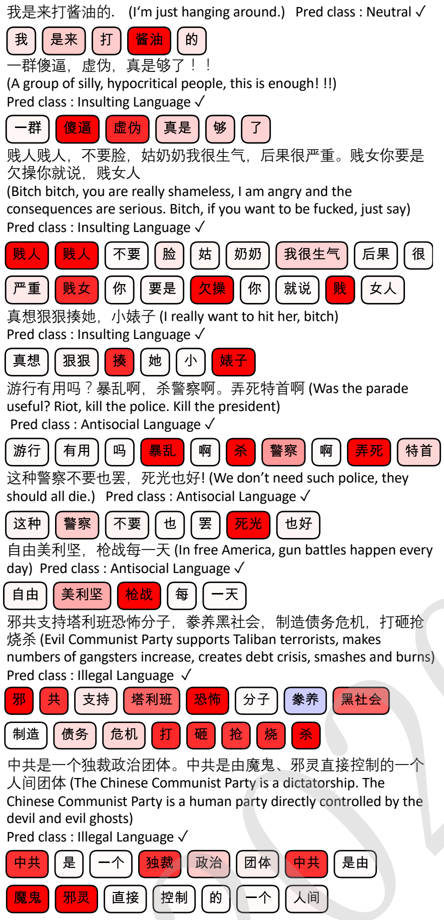
6 Experiments and Results

For the training details, the method of cross-entropy loss was used in CNN, RNN, and BERT. And we used Adam as the optimizer, and the learning rate is set as 0.001. And also, we added dropout and early stop trick. The dropout trick randomly abandons a certain proportion of nodes in the training process to prevent the occurrence of over-fitting. Finally, dropout is Adopted as 0.5. The effects of the model are evaluated by early stop technology on the validation set after each iteration. When the validation set's evaluation result no longer improves in N consecutive rounds, the iterative process is truncated, and the process of training is suspended. The number of N has impacts on the time consumed and Whether the model is converge or not. We implemented our models with PyTorch, moreover, we finetuned the pre-trained based model, BERT, on two NVIDIA RTX 2080Ti GPUs. We also performed an ablation analysis of our HACN model. The weighted F1 drops 1.1% when we remove the hierarchical attention module.

We carry out several experiments during the evaluation phase, and the best experiment is taken into account for the evaluation phase. The systems are evaluated with the official competition metric, Precision, Recall, and Macro Averaged F1 score. What is more, this task is multiple classifications, so macro average, weighted average, micro average are also employed.

In the filtering of insulting language in Chinese, the best performing model achieves a macro averaged F1-score of 94.05%. In the situation of antisocial language, the best performing system for the Chinese achieves a macro averaged F1-score of 97.18%. Finally, in the detection of illegal language, the state of the art NLP model for the Chinese is up to a macro averaged F1-score of 67.30%. However, we adopted

CCL2020



我是来打酱油的.   (I'm just hanging around.)   Pred class : Neutral ✓

一群傻逼，虚伪，真是够了！！
(A group of silly, hypocritical people, this is enough! !!)
Pred class : Insulting Language ✓

贱人贱人，不要脸，姑奶奶我很生气，后果很严重。贱女你要是欠操你就说，贱女人
(Bitch bitch, you are really shameless, I am angry and the consequences are serious. Bitch, if you want to be fucked, just say)
Pred class : Insulting Language ✓

真想狠狠揍她，小婊子 (I really want to hit her, bitch)
Pred class : Insulting Language ✓

游行有用吗？暴乱啊，杀警察啊。弄死特首啊 (Was the parade useful? Riot, kill the police. Kill the president)
Pred class : Antisocial Language ✓

这种警察不要也罢，死光也好! (We don't need such police, they should all die.)   Pred class : Antisocial Language ✓

自由美利坚，枪战每一天 (In free America, gun battles happen every day) Pred class : Antisocial Language ✓

邪共支持塔利班恐怖分子，豢养黑社会，制造债务危机，打砸抢烧杀 (Evil Communist Party supports Taliban terrorists, makes numbers of gangsters increase, creates debt crisis, smashes and burns)
Pred class : Illegal Language ✓

中共是一个独裁政治团体。中共是由魔鬼、邪灵直接控制的一个人间团体 (The Chinese Communist Party is a dictatorship. The Chinese Communist Party is a human party directly controlled by the devil and evil ghosts)
Pred class : Illegal Language ✓

Figure 2: Classification results with explanation tool.

a weighted average as a uniform evaluating method due to the unbalanced classes. Our hierarchical attention capsule network (HACN) is the best-performed model with an excellent result of 94.86%.

7 Analysis

In our experiment, we can find that RNN has an acceptable performance, however, there are some obvious shortcomings. When handling unbalanced data, such as a high recall rate, fail to classify certain classes because the class lacks visible character.

It is also necessary to remark that the experiments are performed with the default parameters. Thus there is an additional field for improvement with some finetuning, which we plan to consider for future research. Moreover, we note that the label distribution is extremely imbalanced because there might be a bias introduced by the algorithms.

Extending from the experiment above, we present a comparison experiment in Figure, where we record the valid accuracy over time and spot trends with different systems. Figure 3 illustrates that our proposed HACN model can quickly converge to its stable equilibrium values. In the meantime, the starting point shows that HACN can get a promising result in a short time (after the first epoch).

Figure 4 shows times spent in the testing step for 3,759 test sentences when using different systems. The curves generated with these results suggested that our proposed HACN model can achieve the classification at the quickest speed. Considering the deployment in a sound system, we only compared the testing step. Nevertheless, we believe we still have substantial advantages in training because there is no need to pretrain the model on a large number of data compared with BERT.

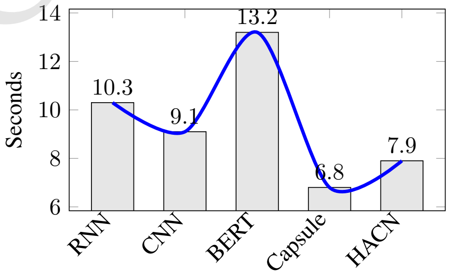
CCL2020

Model	classes	Precision	Recall	F_1
SVM	Neutral	0.7895	0.8211	0.8050
	Insulting	0.8041	0.8444	0.8238
	Antisocial	0.7917	0.2000	0.3193
	Illegal	0.2188	0.1923	0.2047
	Weighted	0.7786	0.7966	0.7824
CNN	Neutral	0.9289	0.9524	**0.9405**
	Insulting	0.9559	0.9657	0.9607
	Antisocial	0.8056	0.9062	**0.8529**
	Illegal	0.4286	0.0380	0.0698
	Macro	0.7797	0.7156	0.7060
	Weighted	0.9270	0.9369	0.9281
	Micro	0.9369	0.9369	0.9369
RNN	Neutral	0.9171	0.9371	0.9270
	Insulting	0.9608	0.9446	0.9526
	Antisocial	0.7120	0.9271	0.8054
	Illegal	0.3415	0.1772	0.2333
	Macro	0.7328	0.7456	0.7296
	Weighted	0.9192	0.9233	0.9202
	Micro	0.9233	0.9233	0.9233
BERT	Neutral	0.9200	0.9459	0.9328
	Insulting	0.9666	0.9771	**0.9718**
	Antisocial	0.7593	0.8542	0.8039
	Illegal	0.7902	0.5861	**0.6730**
	Macro	0.8590	0.8408	0.8454
	Weighted	0.9260	0.9284	0.9259
	Micro	0.9284	0.9284	0.9259
Capsule	Weighted	0.9334	0.9419	0.9376
HACN	Weighted	0.9437	0.9528	**0.9486**

Table 2: Experimental Results and comparisons of our capsule networks and baselines.

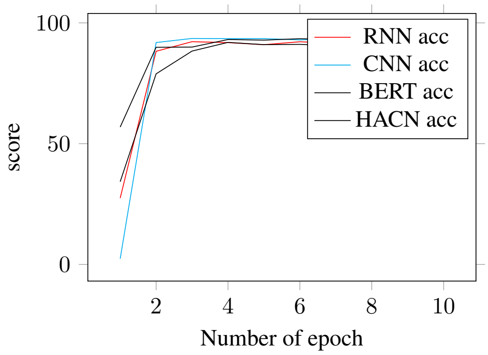
CCL 2020

Figure4: Time costing in testing



Furthermore, We show a deep analysis of the mis-classified cases in the evaluation process of our experiments. Thus, we do a manual analysis for those mis-classified samples. This analysis aims at getting a deep comprehension of the areas our classifiers are lacking in. Our model fails to classify some metaphorical offensive words. It is usual for human to use euphemisms to tone down swear words in some situations. For some other cases, our model classifies some profanity text as offensive, which is actually not offensive. The classifier could miss these word variants, especially when the word variant is the only offensive word in the given sentence. In another situation, the word “sucks” is the only word that is often used offensively. However, the given tweet is not offensive because the author only describes their mood instead of insulting someone else. These misclassifications seem to indicate that the classifier reacts to trigger words with negative connotations but may not be capable of interpreting the words concerning the broader context.

Figure3: Valid Accuracy Performance of each Epoch



8 Conclusion and Future Work

This work presents a Chinese corpus of offensive language crawled from microblog entries and video comments and manually categorized into 4 categories, and several models, including an allegedly novel architecture: Hierarchical Attention Capsule Network, for classification tested on the corpus. we describe the data-set (with simple baselines) and then talk about the modeling with both standard and non-standard methods and tools for explanations. For the dataset, we present an annotated corpus of offensive language in the online world, consisting of sentences and the corresponding annotations. The corpus consists of 18707 sentences annotated with four classes, including neutral language, insulting language, antisocial language, and illegal language. We also present several systems used for the classification of offensive language. The baselines are SVM, RNN, CNN, and BERT. What's more, we present a novel capsule network (HACN) with hierarchical attention to model the semantic structure. The best F1 score of 94.86% is achieved when using HACN. Finally, we propose an explanation tool to illustrate what our systems have learned.

Identifying offensive language in the online world is also interdisciplinary, as it overlaps with psychology, sociology, and economics, while also raising legal and ethical questions, so we expect it to attract a broader audience. Thus, in the future, we would like to bring the ideas and research achievements of other related fields to deliver and share technology and solutions for offensive language from online user-generated content.

CCL2020

Acknowledgements

This research is supported by the National Language Commission Key Research Project (ZDI135-61), the National Natural Science Foundation of China (No.61532008 and 61872157), and the National Science Foundation of China (61572223).

References

Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. 2018. innvestigate neural networks! *arXiv preprint arXiv:1808.04260*.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2017. A unified view of gradient-based attribution methods for deep neural networks. In *NIPS 2017-Workshop on Interpreting, Explaining and Visualizing Deep Learning*. ETH Zurich.

Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2):223–242.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM.

Paula Fortuna, José Ferreira, Luiz Pires, Guilherme Routar, and Sérgio Nunes. 2018. Merging datasets for aggressive text identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 128–139.

Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. 2018. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, page 35. ACM.

Bryce Goodman and Seth Flaxman. 2017. European union regulations on algorithmic decision-making and a âright to explanationâ. *AI Magazine*, 38(3):50–57.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer.

Binxuan Huang and Kathleen M Carley. 2019. Parameterized convolutional neural networks for aspect level sentiment classification. *arXiv preprint arXiv:1909.06276*.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. *arXiv preprint arXiv:1805.01086*.

Karsten Müller and Carlo Schwarz. 2018. Fanning the flames of hate: Social media and hate crime. *Available at SSRN 3082972*.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature visualization. *Distill*, 2(11):e7.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Effective lstms for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. *arXiv preprint arXiv:1805.07043*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

CCL2020

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.

CCL2020