Bully Detection on Weibo

Department of Automation
Shanghai Jiaotong University
@sjtu.edu.cn

Department of Automation
Shanghai Jiaotong University
@sjtu.edu.cn

摘要—This document is an introduction of our Bully language detection project. We will present the purpose and the method to do this project together with the final results. We use several methods to convert Chinese comments into word vector including Word2Vec with TF-IDF, Fasttext, Doc2Vec, Pinyin2Vec using CBOW model. Then, we try traditional machine learning model and use deep-learning model like FCN,GRU,LSTM to get better result. With LSTM and Pinyin2Vec, we got accuracy about 86% to detect the bully comments from Weibo.

***Index Terms*—Word2Vec, Bully-detection, Weibo**

I. Introduction

A. Background

These days, people always surf the internet and using those social media applications had been such a daily routine. However, as giving comments on internet become more and more easy, the quality of those comments have dropped quickly. You can easily see people talk rudely on social media like Weibo, tweet and so on. Sometime, those awful words may not appear but some ironic sentences just appear in someone's message. Those behavior can actually affect others on their heart. These could be called bully language which is as bad as bullying others on the real world.

B. Meaning

Our work is to make our model learn from those bully and non-bully comments and be able to distinguish them. We hope our model can detect those bully sentences and make it hard to appear on the internet which will make the world a better place.

II. Method

A. Word Segmentation

Word segmentation is to divide a sentence into several words with specific semantics, which can reflect the meaning of the sentence as much as possible on the premise of losing some information. The reason is that modeling based on words rather than sentences can improve the generalization ability of the model.

Jieba is a useful python library for Chinese text segmentation. It provides a statistical dictionary contains words and word frequency information. In word segmentation, extract the first corpus in statistical prefix word of each word in the dictionary, such as the prefix of " 结巴分词" includes " 结", " 结巴" and " 结巴分", then look at the prefix word in the dictionary of word frequency statistics. if it does not appear such as " 结巴分", then the word frequency is 0, thus producing the prefix dictionary.

As is shown in Figure1, a **weighted directed acyclic graph** containing all possible word segmentation is formed by taking each word of a sentence as a node and connecting the first and last word of a word that appears in a prefix dictionary to form an edge. A word segmentation scheme corresponds to a participle path from the first word to the last word, and the goal is to find the word segmentation path with the highest probability. The probability of occurrence of each word is defined as the word frequency of the word in the prefix divided by the sum of the word frequency of all words under the prefix, and then take the logarithm (if

the word frequency is 0, the word frequency is regarded as 1).

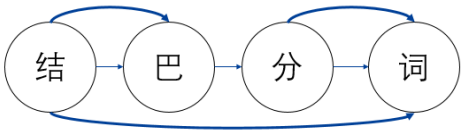

图 1. A directed acyclic graph formed by word Segmentation

Use the **dynamic programming algorithm** to solve the problem. For each node with a degree of 0, it takes out each edge that reaches it, updates the path with the maximum probability, and deletes these edges, and keeps repeating until the edge that can be deleted cannot be found. For words that do not appear in the dictionary, **Viterbi algorithm** and **HMM model** are adopted [1].

B. Word2Vec model

Word embedding is a method of converting words in text into digital vectors, which is easy to analyze using machine learning methods. Compared with the traditional one-hot encoding, word embedding can map the features of words to lower dimensions, reflect the similarity between words, and have strong generalization ability.

Mikolov et al. proposed the **Word2Vec model** [2], including **CBOW** and **skip-gram**. The former is to predict the current word in the knowledge of context, while the latter is to predict the context in the knowledge of current word. In this paper, CBOW model is adopted, which is specifically implemented as a three-layer neural network, including input layer, projection layer and output layer. The network structure is shown in Figure2.

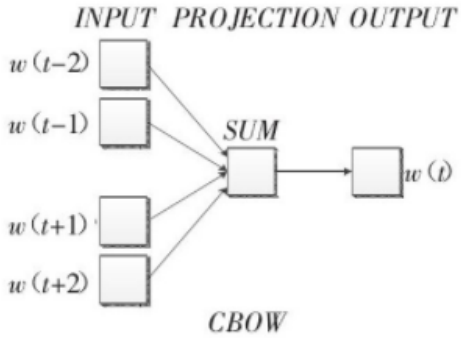

图 2. CBOW model

The model first generates a corresponding vocabulary according to the corpus, and then obtains the context *context(w)*(i.e., n words before and after) of any word *w(t)* in the corpus, so as to build the training sample of the model *(context(w), W)*. The input layer is the word vector of n words before and after the word *w*. The projection is the sum of the word vectors of the 2n words. The output layer is a binary tree constructed with the word frequency of words in the corpus as the weight, and each node corresponds to a vector, including the word vector of leaf node and the auxiliary vector of non-leaf node.

Word2vec is essentially a multi-classification problem corresponding to multiple words, and there are many words in word segment application. Therefore, there are two optimization methods to improve the training efficiency of word vectors, namely **Hierarchy Soft Max** and **Negative Sampling**.**??** Negative Sampling is adopted in this project. Given the sample *(context (w), w)*, *w*is a positive one, and the other words in the vocabulary are negative. After determining a non-empty negative sample set of *NEG (w)*about *context (w)*, for any word in the vocabulary *w'*, give it a tag of 1 if *w'*= *W*and give it a tag of 0 if not. So the positive samples are labeled 1, and the negative samples are labeled 0. The essence of this method is to predict a subset of the total categories and reduce the number of word classifications.

C. TF-IDF

TF-IDF [4],namely term frequency–inverse document frequency, use to assess the importance of a word to one of the documents in a document set or a corpus. The basic idea is that the importance of a word increases proportionally with the frequency of its occurrence in the document, but decreases inversely with the frequency of its occurrence in the corpus.
TF is the frequency of words appearing in the corpus. IDF is the frequency of reverse file, which is a measure of the universal importance of a word. It can be obtained by dividing the number of sentences in the corpus by the number of sentences containing the word, and then taking the quotient as the logarithm base 10, as shown in (1) .

$$IDF(x) = \log \frac{N}{N(x)} \tag{1}$$

Where N is the total amount of text in the text data set, and $N(x)$is the number of text containing the word $x$in the data set. If a word does not appear in the data set, then (1) needs to be smoothed, as (2).

$$IDF(x) = \log \frac{N+1}{N(x)+1} + 1 \tag{2}$$

The product of TF and IDF is TF-IDF value, as shown in (3).

$$w(x) = TF(x) * IDF(x) \tag{3}$$

D. FastText model

Word2Vec takes each word in the corpus as a basic unit, and it generates a vector for each word. This ignores the internal morphological characteristics of words, such as "apple" and "apples", where words have more common characters, that is, their internal morphological information is similar.In traditional Word2Vec, this internal morphological information of words is lost because they are converted to different ids.

To overcome this problem, FastText uses character-level N-grams to represent a word.Take the word "apple" for examle, assume the value of n is 3, then its trigrams are $<$ *ap*, *app*, *ppl*, *ple*, *le* $>$, where $<$ represents the prefix and $>$ represents the suffix. So we can use the sum of these five trigram vectors for the word "apple." This brings two advantages: The word vector for low-frequency word generation works better because their N-gram can be shared with other words.For words outside the training lexicon, it is still possible to construct their word vectors because we can superimpose their character-level N-gram vectors.
FastText is a library for learning of word embeddings and text classification created by Facebook's AI Research (FAIR) lab. [5].As show in Figure 3,the fastText model architecture is very similar to the CBOW model architecture, where x_1,x_2,...,x_N are N-grams features. The features are embedded and averaged to form the hidden variable. The difference between CBOW and FastText is that: the input in CBOW is the context of the target word, while the input in FastText is multiple words and their N-gram characteristics, which are used to represent a single document; CBOW input words have been encoded by one hot, and FastText input features have been embedded ; The output of CBOW is the target term, and the output of FastText is the class label corresponding to the document.In addition, FastText uses hierarchy Softmax in the output, which greatly reduces the model training time.

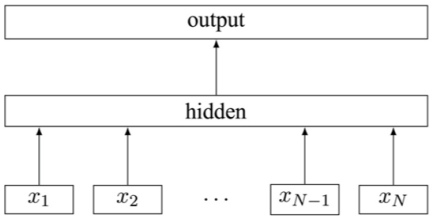

图 3. FastText model

E. Doc2Vec model

Mikolov et al. [6] proposed a **Doc2Vec model** based on the **Word2Vec model**, with the purpose

of directly transforming sentences into vectors, including **PV-DM** model and **PV-DBOW model**. Their main ideas correspond to the CBOW and skP-gram in Word2Vec respectively. Different from Word2Vec model, the input layer includes an additional sentence feature vector, which can be regarded as a vector representing the rest of the information or topic information in the current sentence. The methods of combining sentence feature vector with word vector include mean value and concatenation, as shown in Figure **??** [7].

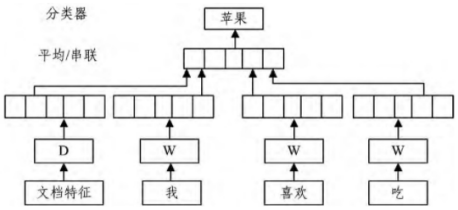

图 4. Doc2Vec model

Sentence feature vectors are used as part of the input each time in the training of several different words of the same sentence. Thus the word vectors are trained, and the theme of the sentence is becoming more and more clear by sharing the same sentence vectors, so as to identify two sentences that are semantically similar, rather than relying on the occurrence of the same vocabulary as TF-IDF algorithms.

F. Pinyin2Vec Model

Pinyin2Vec is proposed for Chinese text analysis, and it is more suitable for social media like Weibo where people often use the word sharing a similar pronunciation with the word the wanted to use in Chinese. In fact, all language can be spoken so this method may be for all language. For example, English words can be indicated by phonetic symbols. And as you know, the data we process and store in the computer must be a string of binary codes, such as UTF-8. Such coding does not reflect the relationship between characters and characters. So Pinyin coding may be more suitable for practical applications.

First, for a word, we take each character as a unit to be converted into pinyin (initial 声母/intermediate vowel 介母/vowel 韵母/tone 声调), and for simplicity, the combination of intermediate vowel and vowel is regarded as one vowel. Then we will find the predesigned position for each initial 声母/intermediate vowel 介母/vowel 韵母/tone 声调, and use the one-hot vector method to encode as the pinyin vector. In this way, 23 initials (if no initials then those position will all be 0) and 36 vowel combinations and 4 tones are encoded for each word which will be a 63-dimensional binary code. In Chinese, we generally use four-character idioms at most, so I decide to accept 4 characters at most, and ignore the redundant part. Then use the CBOW, continuous bag-of-words, method to train the Word2Vec model so that I could use the relationship between the words next to the central word to train the word vectors which could represent the word. And we use the model from input layer to the hidden layer as our pinyin vector to word vector encoding model. With this method, all words in sentence can be first represent by 63-dimension pinyin vector and then converted to 300-dimension word vector which could be used for further classification.

III. Experiment

The main process of the experiment includes data collection and pretreatment, word embedding model construction and machine learning algorithm classification. The workflow is shown in Figure 5.

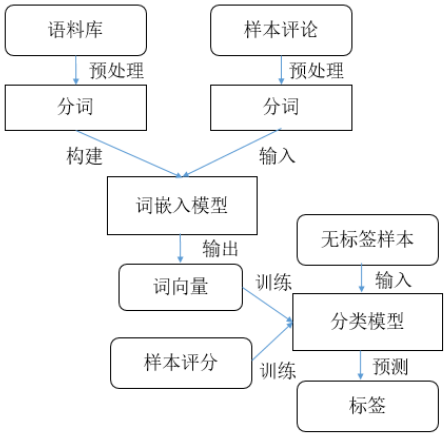

图 5. workflow

A. Data Collection And Pre-treatment

1) Data Source:
   All data are obtained by copying source code of the websites of Weibo and then using **微博数据转换.py** to pick out those Chinese comments.
2) Data Cleaning:
   Invalid comments like " 图片评论 "and comments that is too long to learn the main meaning of it will all be abandoned. By the way, traditional Chinese characters will be converted to simplified Chinese characters and we also have some stop words to be deleted from the sentences. The stop word bank is from HIT.
3) Data Size:
   We have about four thousands comments and they are all labeled by ourselves.

B. Bully language classification

The relevant parameters of the word embedding model are shown in the table I.

表 I
WORD EMBEDDING MODEL PARAMETERS

	w2v	fT	d2v
number	2970	10082	10082
size	300	300	300
workers	4	5	5
min count	1	1	1
window	5	10	10

Among them,"w2v","fT","d2v"represent Word2Vec model, FastText model and Doc2vec model, respectively.*Number* represents the size of the vocabulary built by the word embedding model.*Size*represents the dimension of word vector get from passing a word through the embedding model.*Min count* suggests that words with less than the value of min count are discarded.*Workers* controls the parallel number of training.*Window* represents the maximum distance between the current word and the prediction word in a sentence.

After the word embedding model is established, use different machine learning classifiers to classify, including fully connected Network(FCN),convolutional neural network(CNN),long short-term memory(LSTM),support vector machine(SVM),random forest(RF),logistic regression(LR).The class number is 2,the epoch time is 15,and the learning rate is 0.001.

The structure of FCN is shown in table II.The input dimension is 300.

表 II
FCN STRUCTURE

	type	dimension	activation
layer1	Linear	96	relu
layer2	Dropout	96	None
layer3	Linear	32	relu
layer4	Linear	2	softmax

The structure of CNN is shown in table III.The input dimension is 3*10*10.

表 III
CNN STRUCTURE

	type	dimension	activation
layer1	Conv,BatchNorm,MaxPool	6*4*4	relu
layer2	Conv,BatchNorm,MaxPool	48*1*1	relu
layer3	Flatten,Dropout2d	48	None
layer4	Linear	120	relu
layer5	Linear	2	softmax

The structure of LSTM is shown in table IV.The input dimension is 300. *Num_layers* represents the number of layers of the LSTM hidden layer.

表 IV
LSTM STRUCTURE

	type	dimension	num_layers	activation
layer1	LSTM	96	3	relu
layer2	LSTM	32	3	relu
layer3	LSTM	2	3	relu
layer4	Dropout	2	None	softmax

The accuracy of the classifier is shown in table V.

It can be seen from the results that the accuracy of the classifier is generally around 0.7 and there is still a lot of room for improvement. Therefore, a series of improvement methods are carried out based on this.

表 V
CLASSIFIER ACCURACY

	w2v	fT	d2v
fcn	0.736	0.75	0.769
cnn	0.737	0.739	0.741
lstm	0.739	0.691	0.71
svm	0.739	0.739	0.74
rf	0.768	0.769	0.747
lr	0.683	0.707	0.737

C. Improvement

1) Expand the corpus: The corpus we originally used was the comment information retained after the score was removed from the data set. There were only about 5000 entries in total, and the length of the trained word embedded model vocabulary was less than 10000. We analyze the characteristics of weibo comments and find that they have a wide range of topics and require a large corpus to better understand the semantic features. Therefore, we downloaded a large corpus from the website website , trained the Word2Vec word embedding model, and compared it with the original Word2Vec word embedding model.

The relevant parameters of the word embedding model are shown in the tableVI. *ow2v* represents the Word2Vec model with large corpus.

表 VI
MODEL PARAMETERS

	number	size	workers	min count	window
w2v	2970	300	4	1	5
ow2v	164654	300	10	10	5

The accuracy of the classifier is shown in table VII.

表 VII
CLASSIFIER ACCURACY

	fcn	cnn	lstm	svm	rf	lr
w2v	0.736	0.739	0.739	0.739	0.768	0.683
ow2v	0.803	0.783	0.739	0.779	0.785	0.781

It is obvious that increasing corpus can significantly improve the performance of classifier.

2) Introduce TF-IDF weight: The result of Word2vec model is a word vector, but the sentence needs to be transformed into a vector in the project. We add the converted word vectors after sentence segmentation as the vector of the sentence before, which ignores the role of the word in the sentence, that is, the importance of the word to reflect the meaning of the sentence. Therefore, We introduce TD-IDF value of the words as the weight value, and carry out the weighted sum of each word after sentence segmentation, so that the result is the word vector of the sentence.

Compare the results with those that not introduce TF-IDF, the accuracy of the classifier is shown in table IX.

表 VIII
CLASSIFIER ACCURACY

	fcn	cnn	lstm	svm	rf	lr
ow2v	0.803	0.783	73.891	0.779	0.785	0.781
TF-IDF	0.788	0.785	73.891	0.769	0.794	0.781

It can be known that the result does not change much after the introduction of this value. After analyzing the reasons, IT can be found that weibo comments are often short and lack necessary contextual information. Moreover, the IT-IDF matrix is too sparse and has little impact on the results.

3) Build Pinyin2Vec Model: It can be found that the ow2v model tends to judge bullying based on keywords such as B, lacks a specific understanding of sentence semantics, and fails to identify homophone such as A. Moreover, it has been observed that a large number of weibo comments intentionally use homonyms to achieve bullying due to the blocking of sensitive words. Therefore, we consider using pinyin coding to solve the above problem.

Adopt a multi-layer neural network model similar to CBOW structure to implement word embedding,whose structure is shown in table IX. Be aware that output of layer2 could be regarded as the word vector we want.

We put about five thousands and they were decomposed into over ten thousand short sentences according

表 IX
PINYIN2VEC STRUCTURE

	type	dimension	activation
layer1	Dense	276	tanh
layer2	Dense	300	tanh
layer3	Dense	276	tanh
layer4	Dense	252	tanh

to the punctuation. Then for each short sentence, almost each words in it will be considered as the central keywords and the words next to them with distance less than 2(windows) will be considered as the context and they are a pair of train data. Therefore, we use this unsupervised learning method to learn the relationship between those words and get the word vector.
The accuracy of the classifier is shown in table X.

表 X
CLASSIFIER ACCURACY

	fcn	cnn	lstm	svm	rf	lr
ow2v	0.803	0.783	0.739	0.779	0.785	0.781
py	0.738	0.734	0.739	0.739	0.728	0.745

It is a pity that the final result is not as good as we expected, possibly because the corpus of pinyin model is not large enough.

D. Data Balancing With Negative Sampling

The bully samples are much fewer then non-bully samples. Sometimes it can be found that the accuracy of a certain neural network training is always the same. The actual reason is that the model classifies all the bullying samples as non-bullying samples. In order to balance the proportion of two class of data, we resample the bully samples from the origin datasets to make them almost equal. We ran the experiment on the pinyin2vec model,and the accuracy is shown in figureXI,where *nspy* represents pinyin2cev model with negative sampling.

表 XI
CLASSIFIER ACCURACY

	fcn	cnn	lstm
nspy	0.814	0.78	0.786
py	0.738	0.734	0.739

E. Try new method to convert word vector into sentence vector

It's strange that the LSTM is good at processing and predicting natural language time series but not good at our classification tasks. After analysis, we found that we simply superimpose the word vectors into sentences vectors, and in this process we lose the characteristics of the time series, in other words, the relationship between the word and its context.So we consider stitching up the word vectors in chronological order as input to the LSTM. We also tried another recurrent neural network, GRU, for classification.LSTM model parameters and GRU model parameters are separately shown in the table XII and table XIII. Both in layer1, cover the mask value to make it possible to input variable length data. And both in layer2, return sequences=False.

表 XII
LSTM STRUCTURE

	type	dimension	activation
layer1	Masking	alterable	None
layer2	LSTM	60	tanh
layer3	Dense	30	tanh
layer4	Dense	2	tanh

表 XIII
GRU STRUCTURE

	type	dimension	activation
layer1	Masking	alterable	None
layer2	GRU	60	tanh
layer3	Dropout	60	None
layer3	Dense	10	tanh
layer4	Dense	2	tanh

The results of the LSTM model are shown in figure 6 and figure 7.The results of the GRU model are shown in figure 8 and figure 9.

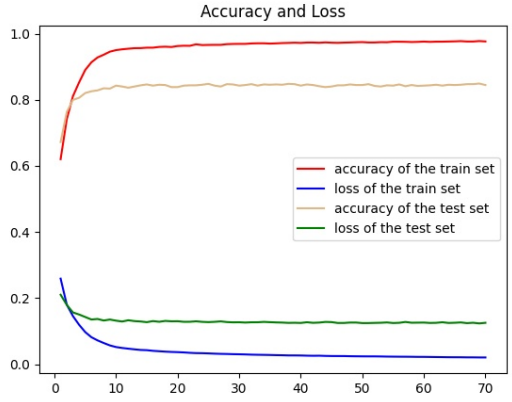

图 6. Loss and Accuracy of LSTM

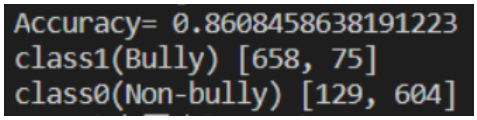

图 7. Confusion matrix of LSTM

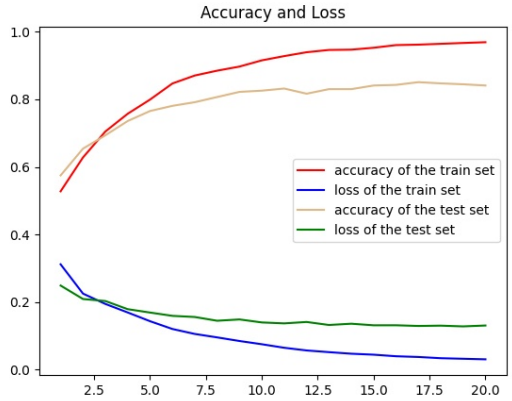

图 8. Loss and Accuracy of GRU

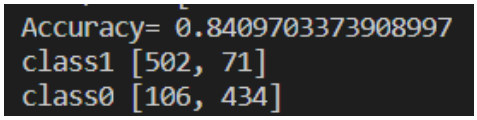

图 9. Confusion matrix of GRU

We can see that when we put into the LSTM and GRU classification network, it just perform quite well compared to the result using pre-trained Word2Vec model. After all, we just have to give each word(Pinyin) a vector that could represent them. So maybe this part has been done quite well, the accuracy could be affected because the network can't predict the context but for a encoder is enough.

Actual input comments are categorized, and the demonstration results are shown in the figure 10.

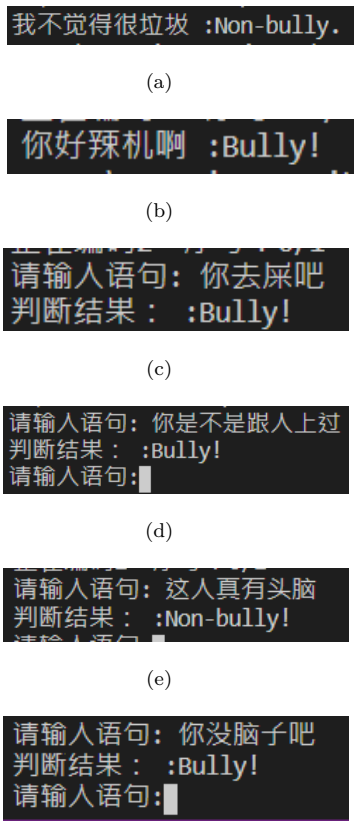

图 10. demo

It can be seen that this model can realize the task of cyberbullying detection well, and can adapt to the classification problems such as harmonics and multi-semantics.

IV. Conclusion

We can see that our model may not be as clever as human but it's efficient and can detect many offensive sentences and some ironic sentences can also be detected.

Bully detection could be such a difficult work to do. First of all, words are not individual, they connected to each other and the order of the words can also affect

the meaning. Second, Chinese words can be much more complicated than English. For example, there are blank space between words in English but there is no easy way to cut the sentences in Chinese so we have to seek help from Jieba and the result may not be that nice because they can't precisely cut the sentence according to their true meaning. Third, people on Weibo don't always use the correct grammar and correct words which forced us to use other methods like Pinyin2Vec to identify the words. Fourth, we don't have enough data and computer resources for word vector training so the word vector may not be that accurate. And we also don't have enough labeled-data for training classification network. For one thing, the data on Weibo may not be that easy to get and it will not be that easy to find enough bully sample because the administer on Weibo may delete or the user will delete if they feel offended. For another, not every share the same principle for bully so some sample will be controversial and it will confused the classification model.

As for the final classification model, we can see from the results that the traditional fully connected network can achieve good results at about 80%. At the same time, the RNN-like models (like LSTM) that considers the front and back connection are quite suitable for our project, which just simulate people's speaking habits and well preserves the context of the text. We people can freely accept and process sentences of different lengths, but for traditional neural networks, they are not very good at them. But using LSTM, each input go through the Masking layer to make the variable length input possible. However, the LSTM may still have some problem to get better result. We consider several reasons. First,we don't have enough data. We can't make bricks without straw so without enough data, the model may not be able to detect some new bully words or some new bully words pattern. And the data don't cover enough area to make the model more robust. Second, the word vector may not be that efficient even we use the Pinyin2Vec. Chinese words are so complicated so it will be so difficult to represent all the words in such small vector of 300-dimensions. Last, there are still some new kinds of networks we haven't try. For example, using a 2D matrix to represent a word then using convolution network like a picture may have better outcomes. Maybe we will continue to improve our work and get better result in the future.

V. Contribution

- Weizhe Kong：Data collecting and cleaning, build pinyin2Vec model and train it using CBOW, build improved FCN,GRU,LSTM network and tuning.
- Weibin Ye：Literature research, data and corpus collection and preprocessing, TF-IDF-weighted Word2Vec model, Doc2Vec model, FastText model,TextCNN model implementation,traditional machine learning classification implementation, build FCN,CNN and LSTM model.

References

参考文献

[1] 邢彪, 根绒切机多吉. 基于 jieba 分词搜索与 SSM 框架的电子商城购物系统 [J]. 信息与电脑 (理论版),2018(07):104-105+108.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, et al.Efficient estimation of word representations in vector space[EB/OL].[2014-09-19].http://arxiv.org/abs/1301.3781v3.

[3] 周练.Word2vec 的工作原理及应用探究 [J]. 科技情报开发与经济,2015,25(02):145-148.

[4] Rong X. word2vec parameter learning explained[J]. arXiv preprint arXiv:1411.2738, 2014.

[5] Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 135-146.

[6] Le Q V,Mikolov T.Distributed Representations of Sentences and Documents[J].Eprint Arxiv,2014,4:1188-1196.

[7] 唐明, 朱磊, 邹显春. 基于 Word2Vec 的一种文档向量表示 [J]. 计算机科学,2016,43(06):214-217+269.