Deep Learning Natural Language Process System for Network Public Opinion Analysis

Weiwei Wang*

Tianjin University Renai College, Tianjin 300000, China

E-mail: 247094015@qq.com

*corresponding author

Keywords: Public Opinion Analysis, Natural Language Processing, Deep Learning, Dynamic Weighting

Abstract: Based on this method, this paper deeply researches natural language processing, based on deep learning, and applies deep learning to natural language processing in order to obtain more accurate results in analyzing the similarity of words in natural language, and combines this method to develop Deep learning natural language processing system for network public opinion analysis. This paper needs to constantly update the existing corpus in the process of processing data, so it is necessary to build a corresponding corpus data server to store real-time corpus data and complete the preliminary processing of corpus data in the database. Segmentation to get data that can be used for deep learning.

1. Introduction

The Ministry of Education held a press conference in February 2019 to introduce the basic situation of the national education development in 2018. In 2018, there were 2,663 ordinary colleges and universities in China, including independent colleges. Among them, 1,245 undergraduate colleges and vocational colleges.There are 1,418 colleges and universities, and 815 postgraduate training units. [1] The total enrollment of various forms of higher education is as high as 38.33 million. With the sharp increase in the number of college students, there is an explosion of online public opinion in colleges and universities, and college students have become the main dissemination body of public opinion in colleges and universities. The increase in the influence of public opinion in colleges and universities not only brings development opportunities for ideological and political work in universities, but also brings impacts and negative effects that cannot be underestimated.

Therefore, how to quickly collect and organize online public opinion information, find out that students are paying attention to hot topics in time, suppress negative emotions, correct wrong public opinions, and reduce the negative impact of emergencies on universities has become one of the important issues facing universities.

Based on the above considerations, a model of sentiment analysis of public opinion in colleges and universities was designed for the post bar websites of major colleges and universities across the country. This model uses web crawlers to collect regularly and uses user comments as the main analysis object. [2]Through natural language processing technology, it discovers hot topics in public opinion. It also analyzes the positive and negative sentiment tendencies of hot topic comments, visualizes the analysis results, and achieves the purpose of analyzing the sentiment tendencies of college public opinion.

2. Model Architecture

The whole model is divided into five stages according to the order of the workflow: corpus preparation, data preprocessing, public opinion monitoring and analysis, performance evaluation, and result visualization.

(1) Corpus preparation: complete the collection of original data and new data through web crawler technology, and complete the sorting of original data at the same time, to prepare for later data training.

(2) Data preprocessing: clean the collected text data, remove dirty data, and complete preparatory work such as word segmentation and part-of-speech tagging through the NLPIR Chinese word segmentation system to prepare for later data feature analysis.

(3) Public opinion monitoring and analysis: Two core function analyses are completed, namely, hot word analysis and sentiment analysis of post content comments.

(4) Performance evaluation: The model sentiment tendency analysis will be evaluated through the accuracy, recall and F value of the classifier.

(5) Visualization of results: Visually display the analyzed data by building a website page.

3. Algorithm Design

3.1 Corpus Preparation

The corpus is divided into two components: original data and new data. The original data mainly plays the role of training set and test set, [3] used to train and test the accuracy of the calculation model, and provide a data basis for selecting sentiment tendency classifiers; the new data is real Evaluate the data to provide a true data source for the visualization stage.

Both types of data are crawled by web crawlers. The difference is that the original data is a one-time crawl with X year X month X day as the demarcation line. The data is classified into comments, divided into positive comment text and negative comment text; and new data will be crawled regularly every day after that day.

The data crawling text mainly contains two, namely baseURL text and id text. The baseURL text is the first crawling text, and the fields include: id serial number), title (post title), link and the topic post comment link); [4] through link Crawl the second id text, as the name suggests, use id as the file name. This id corresponds to the id in baseURL. It is used to save all the information of the id topic post. The fields are: id (consistent with the file name), title (Post title), author (author), time (posting time), floor (comment floor), reviewer (commenter), comment (comment content). Posting time can be used to sort the latest posts later, and how many comment floors can be used for popular posts for sorting, the author and commenter can be used to count the person's activity in the later stage, and the comment content is the basic data for the analysis of public opinion sentiment tendency.

3.2 Data Pre-processing

First, clean the crawled text data to remove dirty data such as incomplete information, emoticons, and garbled text. For the processed text data set, NLPIR (full name: NLPIRICTCLAS Chinese word segmentation system) is used to segment the text. The system is developed by the Chinese Academy of Sciences team and has the functions of segmentation and part-of-speech tagging for Chinese and English information. Prepare two tables before word segmentation: Stop word list and user-defined table, by removing stop words and adding custom user words to improve the accuracy and pertinence of word segmentation. Stop word list is mainly used to filter some invalid, meaningless or disturbing words, such as Words, words or phrases such as "ah", "oops", "Not only"; the user-defined table is mainly used to set up some subject vocabulary and special online vocabulary, such as "Beida", "Guoke", "Geely" and "True Xiang" Since this model is used to process public opinion in universities and other unconventional phrases, custom user words will be added for universities as the object, such as "high numbers", "big things", "online lessons", etc. The text after data segmentation is defined as comments the word comments_ words text, which is used for hot word analysis and related word analysis in public opinion monitoring and analysis.

3.3 Public Opinion Monitoring and Analysis

Public opinion monitoring and analysis is the core part of the entire model. The following is a

detailed description of hot word analysis and comment sentiment analysis. [5]

3.3.1 Hot Word Analysis

LDA is a relatively mature document topic generation model. Its essence is a three-layer Bayesian probability model, [6] which is often used in natural language processing. Considering the limitations of the traditional LDA topic model, this part is integrated into the traditional LDA topic model. The category TF-IDF algorithm realizes the keyword extraction of the text category.

The TF-IDF algorithm is a combination of the TF algorithm and the IDF algorithm. It is a classic data mining feature weighting algorithm. The TF algorithm is used to obtain the number of times a word a appears in the document J, that is, the word frequency tf(i,j),n(i,j) represents the number of occurrences of word a in document J, n(k,j) represents that there are k words in document j, then tf(i,j) is expressed as:

$$\mathrm{tf}(\mathrm{i},\mathrm{j}) = \frac{\mathrm{n}(\mathrm{i},\mathrm{j})}{\sum_{\mathrm{k}} \mathrm{n}(\mathrm{k},\mathrm{j})} \tag{1}$$

Considering that some words may appear frequently in documents, but their importance is often not high, such as words such as "excuse me", "things", etc., so the IDF algorithm can be used to obtain the weight of the universal importance of a word i in the document. , That is, the inverse document term frequency idf(i), where |D| represents the total number of documents in the corpus, and dj represents the number of documents where the word i appears in the corpus, then idf(i) can be expressed as:

$$\mathrm{idf}(\mathrm{i}) = \lg \frac{|\mathrm{D}|}{|\{\mathrm{j}:\mathrm{t}_\mathrm{j} \in \mathrm{d}_\mathrm{j}\}|} \tag{2}$$

Then TF-IDF is that the greater the weight obtained by multiplying tf(i,j) and idf(i), the higher the probability of the word becoming a keyword, expressed as:

$$\mathrm{tf}(\mathrm{i},\mathrm{j}) * \mathrm{idf}(\mathrm{i}) = \frac{\mathrm{n}(\mathrm{i},\mathrm{j})}{\sum_{\mathrm{k}} \mathrm{n}(\mathrm{k},\mathrm{j})} \lg \frac{|\mathrm{D}|}{|\{\mathrm{j}:\mathrm{t}_\mathrm{j} \in \mathrm{d}_\mathrm{j}\}|} \tag{3}$$

On the basis of obtaining the weight matrix of each word, construct the LDA model object. The LDA model believes that a certain topic can be reflected through several words, and a document can contain multiple topics, so the process of generating a document includes two Stage: From words to topics, and then from topics to documents.

Let w represent words, t represent topics, and d represent documents. Let y represent the number of words corresponding to the i-th topic in the document, and n represent the number of words in the document, then the i-th topic t in document d can be expressed as:

$$\mathrm{tf}(\mathrm{i},\mathrm{j}) = \frac{\mathrm{n}_{\mathrm{ij}}}{\mathrm{n}} \tag{4}$$

Taking the comment data of a certain day as an example, the entire implementation process is as follows:

(1) Load the comments words file that has been processed by word segmentation;

(2) Construct the TfidfVectorize: object, perform multiple experiment adjustments to find the maximum eigenvalue, and obtain the TF-IDF weight matrix of the comment text through the fit_transform function.

(3) On the basis of obtaining the weight matrix of each word, construct the LDA model object, obtain TopN hot words by setting the LDA model object parameters, and use the fit function to complete the data set training of the weight matrix of each word, and finally train two A result vector, namely P(wlt) and P(tld) two probability distributions.

(4) Obtain the final weight data of the entire hot word through the get_feature_name and function in the TfidfVectorize: object, and sort it in descending order according to the weight to obtain the hot word list.

3.3.2 Emotional Tendency Analysis

The selection of sentiment classifier is a very important part of the realization of sentiment tendency analysis function. This part needs to train the original corpus data and complete the model evaluation through the test results of the test data set. [7]

First, classify the original data according to the positive and negative sentiment of the comments, sort out the positive and negative texts, and accumulate about 60,000 comments in the corpus. Use NLPIR to segment the data sets in these two texts and remove the stop vocabulary list. Add user-defined vocabulary, set feature value 1 for positive vocabulary, and set feature value for negative vocabulary. , Realize the classification of two data sets: the positive vocabulary list po and the negative vocabulary list neg.

Secondly, in order to increase the accuracy of classifier selection, the data arrangement of the two data sets is randomized and then combined, and the combined data set is randomly divided into training set x_train and test set x_tPSt.

Then proceed to the classifier selection. The data set will be divided into three scales, and 4 classifiers will be tried, namely: B Bernoulli Bayes classifier, discrete naive Bayes classifier, logistic regression classifier and support vector Machine linear classifier. In order to measure the accuracy and practicability of different classifiers, three evaluation indicators are used to describe the test effect, namely: accuracy, [8] recall and F value, and the results are calculated and summarized.

Among the four classifiers, the logistic regression classifier and the support vector machine linear classifier performed better, while the logistic regression classifier performed relatively better.

Taking a piece of post comment data as an example, the entire implementation process is as follows:

(1) Construct a linear regression classifier, construct a training set based on the 60,000 original data in the corpus, and complete the linear regression classifier training through the training set.

(2) Complete the collection and preprocessing of a certain post data to form a new post comment text. The fields include the post id and all comments under the id. Read the comment text of the post and construct all the comments under the post X .

(3) Construct a new set of comments Y, iterate out each sentence of comments from the old set of comments X, generate a bag of words D, that is, a dictionary, D represents the bag of words of the ath comment, and perform word segmentation for each sentence of comments. Need to remove the stop words and add custom user words, mark each valid word after word segmentation, put it into the word bag, and then put D into Y.[9]

(4) Classify the new comment set Y through the trained classifier, and return the entire comment classification result set. The positive classification is 1, and the negative classification is 0. Assume that the number of positives in the set is P and the number of negatives is Nm, Then the proportion of positive emotions r(p) and the proportion of negative emotions r(n) in the post comments can be expressed as:

$$r(p) = \frac{P_n}{P_n + N_m} * 100\% \quad (5)$$

$$r(n) = \frac{N_m}{P+N} * 100\% \quad (6)$$

(5) Visualize the results of the post information and the emotional tendency of the comments under the topic post.

4. Conclusion and Suggestion

This model uses natural language processing technology to analyze the sentiment tendency of comments on the topic of university public opinion. Since this model is mainly used to realize the connection of the entire hot word analysis and comment sentiment analysis process, there is no preliminary requirement for the amount of data, and the later can Significantly increase the amount of public opinion data. By adopting Hadoop-related technologies to realize the distributed storage and data processing of large data sets, the speed of preliminary data preparation [10] and

preprocessing is greatly reduced, and a Hadoop-based university post bar public opinion monitoring system is realized in a true sense. And this model does not involve public opinion information contained in other carriers such as pictures and emoticons in comments, and further research is needed.

References

[1] Peiguang Li,Hongfeng Yu,Wenkai Zhang,Guangluan Xu,Xian Sun. SA-NLI: A Supervised Attention based framework for Natural Language Inference. Neurocomputing,2020,407.

[2] Zahra Rahimi,Mohammad Mehdi Homayounpour. Tens-embedding: A Tensor-based document embedding method. Expert Systems With Applications,2020,162.

[3] Wasan AlKhwiter,Nora Al-Twairesh. Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM. Computer Speech & Language,2021,65.

[4] Thomas Belligh,Klaas Willems. What's in a code? The code-inference distinction in Neo-Gricean Pragmatics, Relevance Theory, and Integral Linguistics. Language Sciences,2021,83.

[5] Sanggyu Chong,Sangwon Lee,Baekjun Kim,Jihan Kim. Applications of machine learning in metal-organic frameworks. Coordination Chemistry Reviews,2020,423.

[6]Cho Insook,Lee Minyoung,Kim Yeonjin. What are the main patient safety concerns of healthcare stakeholders: a mixed-method study of Web-based text.. International journal of medical informatics,2020,140.

[7] Tran Duc Chung. The First Vietnamese FOSD-Tacotron-2-based Text-to-Speech Model Dataset.. Data in brief,2020,31.

[8] Rezaei Zahra,Ebrahimpour-Komleh Hossein,Eslami Behnaz,Chavoshinejad Ramyar,Totonchi Mehdi. Adverse Drug Reaction Detection in Social Media by Deepm Learning Methods.. Cell journal,2020,22(3).

[9] Emily Bell,Tyler A. Scott. Common institutional design, divergent results: A comparative case study of collaborative governance platforms for regional water planning. Environmental Science and Policy,2020,111.

[10] Hao Fei,Yafeng Ren,Donghong Ji. Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction. Information Processing and Management,2020,57(6).