2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Natural Language Classification Algorithm of Comments Based on Bayesian Chasing-Clustering Model

Yifeng Wang, Yang Wang, Weisheng Chen, Yingzhen Lin
School of Science, Harbin Institute of Technology
Shenzhen, China
wangyifeng_ai@163.com

***Abstract*—The clustering algorithm groups data objects according to their similarity. As an unsupervised learning, it is very convenient to use because there is no need to set labels for the data before training. But at the same time its clustering results are not attached with corresponding grouping information, which means the specific meaning of each type of data. However, in the field of modern Internet, the dimension of information space is very high, especially for user comment sections involving natural language. If the clustering direction is not induced and restricted, it is easy to appear that the clustering results are far from the goal of the algorithm. Therefore, we propose a Bayesian chasing-clustering model, which is improved on the basis of the k-means clustering algorithm. During the training process, it will tend to the clustering way we set in advance, and finally achieve a clustering effect which meets our goals better. In addition, we also propose the S-T method for word embedding representation, which is very suitable for natural language processing problems about comments. We applied it to film review websites and e-commerce platforms, and clustered the website content based on users' comments. Compared with traditional clustering methods, the accuracy rate was improved by 9%-57%.**

Keywords-chasing-clustering; e-commerce platform; S-T method; comment-type natural language;

I. INTRODUCTION

In recent years, the Internet field has developed rapidly. And comment sections are an important part of e-commerce, social platforms and various of Q&A or content sharing communities. Users can get a general understanding of relevant content through the comment area. For example, through the movie comments, users can get a general understanding of the movie's quality, category, theme and even the values that are promoted, and then gain some basis for selection. Taking the e-commerce platform as another example, users can check the comments of different products to gain a deeper understanding of the product, which is also an action that users often take when shopping online. However, the number of comments is always so large that consumers are overwhelmed. What's more, there is also a phenomenon of default reviews and merchants' reviews. All these phenomena make consumers feel bad when shopping. With the application of clustering, the comments can be automatically summarized according to the similarity between them. Therefore, a win-win situation can be achieved for both consumers and merchants by collecting relevant data from the e-commerce platform, constructing appropriate feature vectors for the same kind of commodities of different merchants, and then implementing cluster analysis of the commodities.

For the problem of natural language processing, the first problem we face is to represent the text with vectors. In 2013, the Google team published the word2vec tool, which is a commonly used model tool for word embedding training [2]. The common practice is to use the word2vec method to pre-train word embeddings on the public data set, and then load the trained word embeddings into your own model, fine-tune them, and adjust them into word embeddings which are suitable for your own data set. Compared with the traditional vector space model, the word2vec model can map feature words to specified dimensions, greatly reducing the dimension space [3]. The word2vec tool mainly contains two models [4]: skip-gram model and Continuous Bag of Words(CBOW). These two methods are to train word embeddings through the emergence of context, and finally some parameters of the model are used as word embeddings [5]. These word-embedding representation methods have excellent universality. However, they usually lack of special effects for some special types of natural language data, especially for our comment-type natural language data [6]. Therefore, according to the characteristics of comment-type natural language, we have improved the representation method of word embeddings, and proposed an S-T model suitable for comment-type natural language.

In addition, when using traditional cluster algorithms to cluster comment-type natural language data, there are often cases where the clustering results are far from our expected goals [7-9]. For example, we want to cluster the quality of goods, but the result is the types of them. Therefore, we have improved the cluster algorithm and designed a chasing clustering algorithm based on Bayesian theory [10-12], which can effectively solve this problem.

II. S-T METHOD FOR COMMENT-TYPE NATURAL LANGUAGE

A. Construction of S-T method

In the vector-expression of feature words, we should pay attention to the data characteristics of the problem we research, and comment-type natural language often have the following characteristics:

1) The repetition rate of feature words is high.

2) Compared with ordinary languages, comment-type natural language has a smaller coverage of common words.

3) Different types of objects, movies, music, novels, etc., have very different comment areas. For example, we can use "moving to the deep" to describe literary and artistic works, but it is very inappropriate to use it to describe an object. As another example, we can use "beautiful and fit" to describe a piece of clothes, "sweet and delicious" to describe a food, and "high configuration and speed" to describe a server, etc., but they are obviously not arbitrary. In other words, for different types of items, comments on their contents have obvious differences and boundaries.

Based on the characteristics above, we propose a S-T model suitable for comment-type natural language on the basis of the two word vector representation models of TF-IDF and skip-gram, and use the movie review data set to conduct a clustering-comparison test on the S-T model and other word embedding representation methods.

The specific construction method of the S-T model is described by taking "Harbin Institute of Technology is a famous university in the world" as an example.

First, randomly initialize and create two matrices—an embedding matrix and a text matrix. These two matrices embed all words in our vocabulary, as shown in Fig. 1.

Set up a word prediction task and train a deep learning model to complete the probabilistic prediction of the subsequent content of the input words, as shown in Fig. 2.

After training, the model will generate a matrix containing all the words in the mapping word list. When making predictions, the function of our algorithm is to query the input words in this mapping matrix, and then calculate the predicted value.

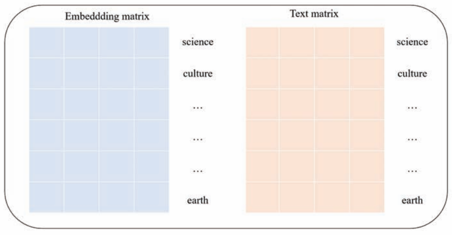


Figure 1. Two randomly created matrices.

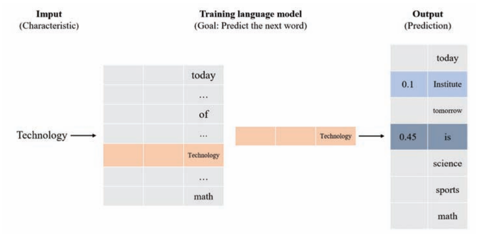
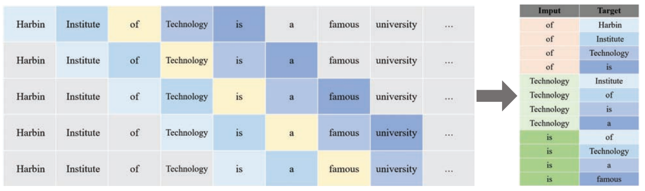


Figure 2. Setting the natural language prediction task.

At this point we need to generate a data set for model training. The specific method is to set up a sliding window and slide along the corpus from beginning to end. During the training, different sliding window lengths can be set according to requirements. The length of the sliding window is usually set to 5, and there are two related words before and after each central word (except the beginning and end). We use the center word as the training input of the model. The center word of the sliding window is marked in red. Different related words have different weights. Therefore, they are expressed in different colors. After moving several groups of positions, we can get a batch of samples, as shown in Fig. 3.

Calculate the dot product of the embedding vector of the input word and the embedding vector of the context word. The result is the similarity of the input word and the context word. Then convert the similarity into a positive value between 0 and 1 by the sigmoid function.



Harbin Institute of Technology is a famous university …
Harbin Institute of Technology is a famous university …
Harbin Institute of Technology is a famous university …
Harbin Institute of Technology is a famous university …
Harbin Institute of Technology is a famous university …

Imput	Target
of	Harbin
of	Institute
of	Technology
of	is
Technology	Institute
Technology	of
Technology	is
Technology	a
is	of
is	Technology
is	a
is	famous

Figure 3. Schematic diagram of training samples obtained by sliding window.

TABLE I. CALCULATE THE DOT PRODUCT OF INPUT WORDS AND OUTPUT WORDS

Input word	*Output word*	*target*	*Input output*	*Sigmoid ( )*
Technology	Institute	0	0.74	0.68
Technology	of	0	-1.11	0.25
Technology	is	1	0.20	0.55

According to the function value of 'error value = target value-sigmoid', the predicted value is compared with the real target label to calculate the model error:

TABLE II. CALCULATE THE DOT PRODUCT ERROR VALUE OF INPUT WORDS AND OUTPUT WORDS

Input word	*Output word*	*target*	*Input output*	*Sigmoid ( )*	*error*
Technology	Institute	0	0.74	0.68	-0.68
Technology	of	0	-1.11	0.25	-0.25
Technology	is	1	0.20	0.55	0.45

Using the results of the error function as the updated values of the model embedment function. As shown in Fig. 4, loop through the entire data set multiple times, and the embedding will continue to be improved. Then we can stop the training process, discard the text matrix, and use the embedding matrix as the trained embedding for the next task.

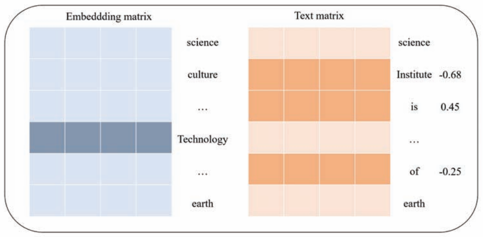


Figure 4. Get the updated function value of the model.

Calculate the TF-IDF value of each feature word according to the comment-type natural language data set we collected. After getting the TF-IDF vectors of all feature words, calculate the dot product of them and the matrix in the obtained training model. Finally we get the word embeddings of the comment-type natural language feature words, as shown in Fig. 5. Hereinafter referred to as S-T model.

The S-T method combines the advantages of the skip-gram model and the TF-IDF model. Besides focusing on the context semantic information of the feature word, it also pays attention to the importance of the feature word to the specific task, which is very important for comment-type natural language data. Compared with ordinary natural language data, this kind of data is more fragmented by different types of evaluated things, and the scenes of the same feature word are also relatively limited, so there is no need to give it complicated semantic information at most time. But at the same time, the comment-type natural language is extremely concerned about some special feature words. The S-T model injects such attention into the expression of word embeddings, so it will have a better expression effect.

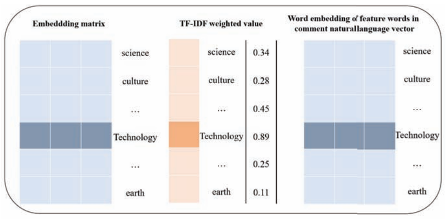


Figure 5. The S-T model expresses the word embeddings of comment-type natural language.

B. *Clustering of movie reviews based on S-T method*

To effectively represent movie reviews by using word embeddings is a current difficulty, and the current pace of research in this area is slow. A common method is to average all word embeddings contained in the short text, but this method ignores the importance of a single word embedding to the text representation, making the representation of the short text embeddings less accurate. The process of using word embeddings to express the texts of movie comments is as follows:

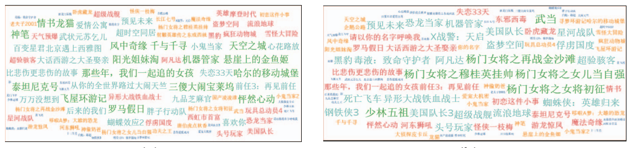
1) There are five major types of movies, including love, comedy, fantasy, cartoon, and swordsman. Select 20 representative films from each of these types for a total of 100 films. Crawl the top 100 comments of each movie's hot review as its comment document, and assume the size of the test corpus as D_n at this time.

2) Use "jieba" to segment each movie comment, and use the Chinese stop word list published by the Chinese Natural Language Processing Open Platform of the Institute of Computing Technology of the Chinese Academy of Sciences. Remove stop words, irregular words, etc., and retain nouns, verbs, adverbs, idioms and other 8 types of parts of speech words, and finally get n candidate keywords: $D = [t_1, t_2, \dots, t_n]$.

3) Because wiki Chinese corpus is recognized as a large Chinese corpus, we use it as the basic corpus for the construction of word2vec model. The main steps are to obtain the Wikipedia Chinese corpus, to extract plain text, data preprocessing, and jieba word segmentation processing. After processing, 1021349720 original words are obtained, and then 952969230 effective words obtained after training become the final word embeddings.

4) All candidate keywords in each movie comment are weighted. Calculate the word frequency TF of the word in the text D, then calculate the inverse document frequency IDF in the entire corpus, calculate the TF-IDF value of all candidate words, and finally we leave the top 50 vocabulary from each movie review document as keywords of the movie.

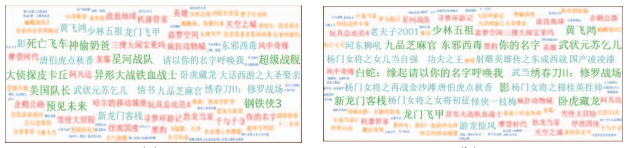
After the vector representation of the movie with word2vec and S-T methods, the resulting embeddings are used as the input embeddings for clustering. Using both the traditional word2vec method and k-means cluster algorithm to perform clustering, and the results are shown in Fig. 6(a). The improved S-T model is combined with the k-means clustering algorithm for clustering, and the results are shown in Fig. 6(b).



(a) (b)

Figure 6. K-means clustering result of traditional word2vec model (a) and the improved S-T model (b).

Then the same movie representation embeddings are used as the input embeddings of spectral clustering, and the traditional word2vec method and S-T method are combined with the spectral clustering algorithm respectively, and then the same 100 movie review text sets are clustered. The obtained clustering results are shown in Fig. 7(a) and Fig. 7(b) respectively.



(a) (b)

Figure 7. Spectral clustering results of traditional word2vec model (a) and the improved S-T model (b).

Use F_1 to compare the significance results of the clustering results before and after improvement, and the results are shown below.

As shown in Fig. 8(a) and Fig. 8(b), when using the ST model proposed in this paper for text feature extraction, the saliency evaluation index value of the movie review data set clustered, F_1, is higher than the traditional word2vec model, whether the combined clustering algorithm is k-means clustering or spectral clustering. Especially, the F_1 of the cluster algorithm after combining the S-T model with k-means is significantly higher than the F_1 of the traditional word2vec model.

TABLE III. ACCURATE VALUES OF K-MEANS CLUSTERING BASED ON WORD2VEC AND S-T MODEL

Category	***word2vec model***	***S-T model***
Love	50.00%	78.68%
Comedy	34.82%	41.95%
Fantasy	57.91%	73.12%
Cartoon	47.10%	61.13%
Swordsman	47.39%	90.00%

TABLE IV. RECALL RATE OF K-MEANS CLUSTERING BASED ON WORD2VEC AND S-T MODEL

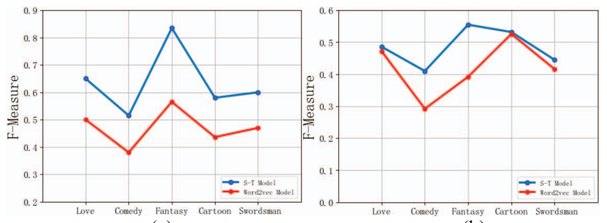
Category	***word2vec model***	***S-T model***
Love	50.00%	55.00%
Comedy	40.00%	65.00%
Fantasy	55.00%	95.00%
Cartoon	40.00%	55.00%
Swordsman	45.00%	45.00%

TABLE V. K-MEANS CLUSTERING VALUES OF WORD2VEC MODEL AND S-T MODEL

Category	***word2vec model***	***S-T model***
Love	50.00%	64.70%
Comedy	37.20%	51.12%
Fantasy	56.40%	82.60%
Cartoon	43.20%	57.96%
Swordsman	46.20%	60.00%

TABLE VI. SPECTRAL CLUSTERING VALUES OF WORD2VEC MODEL AND S-T MODEL

Category	***word2vec model***	***S-T model***
Love	46.80%	48.30%
Comedy	28.62%	41.00%
Fantasy	39.10%	55.00%
Cartoon	52.20%	53.00%
Swordsman	41.20%	44.00%



(a) (b)

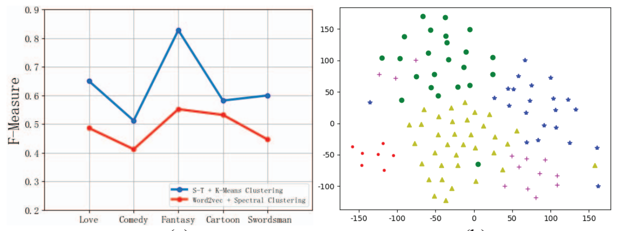
Figure 8. Significance evaluation index value (F_1) of k-means clustering of word2vec and S-T model (a) and spectral clustering of word2vec and S-T model (b).

Combine the improved S-T model with k-means and spectral clustering respectively, and compare the results of the significant evaluation indicators (F_1). The experimental results are shown in Fig. 9(a).

As can be seen from Fig. 9(a), the S-T feature extraction model under the k-means cluster algorithm has a better evaluation index value (F_1) than spectral clustering. Therefore, it can be concluded that the k-means cluster algorithm is more suitable for the clustering of text data sets after high-dimensional, and multi-text feature extraction.

The clustering results obtained by k-means are visualized by using the t-SNE dimensionality reduction method. The results are shown in Fig. 9(b).

It can be seen from the visualized results after dimensionality reduction of 100 movies that the algorithm still has a lot of room for improvement. Although we can roughly distinguish the categories of different movies, there are still some samples that are clustered into the wrong category. The boundaries between the categories are not clear, and the specific nature of different categories is difficult to reflect. All of them reflect the ambiguity of clustering standard. At the same time, although the significance evaluation index value (F_1) of the improved S-T model is better than the F_1 of the unimproved word2vec model, the clustering effect of "high cohesion, low coupling" is not obvious, and the difference between the clustering results is not obvious, too. Therefore, we constructed a Bayesian chasing clustering model.



(a) (b)

Figure 9. (a) S-T model's significance evaluation index value (F_1) in k-means and spectral clustering. (b) K-means clustering results of 100 movies.

III. BAYESIAN CHASING CLUSTERING MODEL

Although the S-T model is better than the traditional word embedding representation model for comment-type natural language clustering, the significance of the movie comment data set still needs to be improved. Therefore, based on the Bayesian idea, this paper proposes a chasing clustering model. On the basis of the k-means model, the conditional probability of the feature item is changed, and the constraint conditions are added when clustering each document. Movie comments are still selected as the data set, and the S-T method is used as the word embedding representation method. The experimental results prove that the clustering effect of the improved chasing clustering model is better than that of the k-means model.

A. Construction of chasing clustering model

The main idea of establishing a "chasing clustering" model is that before clustering, feature items are mainly calculated according to the distance or similarity coefficient between them, and it is not known which categories the whole will be divided into. Therefore, based on the Bayesian idea, we hope to enhance the directionality of the data sample clustering process by changing a small constraint. Now, we call the process of changing the constraint "chasing clustering" process. For the comment text data set, we increase the conditional probability of the feature items in the data in the text. The modified feature words are called the headwords of chasing clustering.

The main feature of Bayesian method is to derive a posteriori knowledge through prior information. Therefore, when clustering is based on the Bayesian method, we need to assume that the data in the data set that may be clustered into a class follow a certain distribution based on the priori information. Then use the Euclidean distance to detect whether the distribution given by the prior information meets the requirements of clustering. If the clustering requirements cannot be met, you need to recalculate the probability, find the reason for not meeting the clustering requirements, and determine the distribution again or modify the parameters of the distribution.

The document x is expressed in the form of a vector $X(x_1, x_2, \ldots, x_n)$, and by using Bayes' formula, the aggregate of categories $C(c_1, c_2, \ldots, c_n)$ can be written as:

$$P(c_k|x_1, x_2, \ldots x_n) = \frac{P(x_1, x_2, \ldots x_n|c_k)P(c_k)}{P(c_1, c_2 \ldots c_k)}) \qquad (1)$$

$P(x_1, x_2, \ldots x_n|c_k)$ represents the probability that the document x belongs to the category c_k when it contains the vector $X(x_1, x_2, \ldots, x_n)$. $P(c_k)$is the a priori probability that the document x belongs to the category c_k , and $P(c_1, c_2, \ldots, c_k)$ is the joint probability of all given categories.

Since the denominator is the same for the joint probability calculated of all categories, the categories to which the document belongs can be clustered according to the maximum value of the calculated posterior probability. Rewrite (1) by using argmaxfunction to seek the aggregate of events which have the maximum probability to happen.

$$c(x) = \underset{c_k \in C}{argmax}\, P(x_1, x_2, \ldots x_n|c_k)P(c_k) \qquad (2)$$

In practical applications, in order to reduce the complexity of Bayesian network construction, we assume that the attributes of data are independent of each other. Then the Naive Bayes formula is：

$$c(x) = \underset{c_k \in C}{argmax} P(c_k) \prod_{i=1}^{n} P(x_i|c_k) \qquad (3)$$

The priori probability $P(c_k)$ and conditional probability $P(x_i|c_k)$ are expressed as follows:

$$P(c_k) = \frac{\sum_{j=1}^{n} \delta(c_j, c_k)}{n + l} \qquad (4)$$

$$P(x_i|c_k) = \frac{\sum_{j=1}^{n} \delta(x_{ji}, x_j)\delta(c_j, c_k) + 1}{\sum_{j=1}^{n} \delta(c_j, c_k) + n_i} \qquad (5)$$

In (4) and (5), n is the number of training samples, l is the number of categories, n_i is the number of the i. feature value, c_j is the category of the j. training sample, a_{ji} represents the i. feature value of the j. training sample. $\delta(\cdot)$ is a binary indicative function. When the parameters are the same, its value is 1, otherwise it is 0.

On the basis of the S-T model proposed above, the feature items are given different weights for calculation according to the degree of predicted feature items' dependence on other feature items and adhering to the principle of "high dependency and low weight". Add weight constraint conditions in the conditional probability formula, and add the weight of the selected feature on the basis of (3):

$$c(x) = \underset{c_k \in C}{argmax} P(c_k) \prod_{i=1}^{n} P(w_i x_i|c_k) \qquad (6)$$

The parameter ω_i is the weight of the i. Feature x_i. After adding the constraints, the prior probability formula remains unchanged, but the conditional probability (5) changes into:

$$P(w_i x_i|c_k) = \frac{\sum_{j=1}^{n} w_i \delta(x_{ji}, x_j)\delta(c_j, c_k) + 1}{\sum_{j=1}^{n} w_i \delta(c_j, c_k) + n_i} \qquad (7)$$

B. *Algorithm implementation of chasing clustering model*

Based on the construction of the chasing clustering model, we have designed the corresponding implementation algorithms. The specific process of the algorithm is shown in Table VII.

In the process of algorithm implementation, we need to calculate the TF-IDF weight values of different documents. Based on the calculation results, some clustering centers selected artificially can effectively determine the clustering direction for the model, so that the clustering results of the model and our the starting point are basically the same.

TABLE VII. ALGORITHM STEPS OF CHASING CLUSTERING MODEL

Input:	The determined center chasing point, the representation vector of the characteristic word embeddings of the S-T model
Output:	The clustering results
Step1:	Select the best center chasing point of the whole space
Step2:	Distribute high weight to documents including x_i, and low weights to others according to the "high dependency, low weight" idea.
Step3:	For each category that has determined its center chasing point, calculate $P(c_k)$ as the prior probability according to the prior information.
Step4:	Add center chasing point calculation, use k-means cluster algorithm for clustering, then obtain each c_k and the posterior probability after data clustering calculation.
Step5:	Check the clustering result. If it meets the requirements, the algorithm ends, and output the clustering result. Otherwise re-determine the distribution or modify the parameters.

IV. CLUSTERING APPLICATION BASED ON CHASING CLUSTERING ALGORITHM

A. *Clustering of community comments targeting movie categories*

In the movie comment community, each movie is accompanied by a large number of user comments, which cover the type, quality, and value system of the movie. We need the model to focus on the type information of the movie and use it as the basis to accomplish clustering.

The experimental data is still the 100 movies' comment data previously obtained from Douban.com. First, we need to calculate the TF-IDF value of each feature word in the data set and artificially specify some chasing center points, as shown in Table VIII:

Count the frequency of chasing center points appearing in 100 movies and sort them in order from high frequency to low frequency. The result is: love, cartoon, comedy, fantasy, swordsman. According to the principle of "high dependence, low weight", increase the weight of selected features and distribute different weight values to different chasing center points. After multiple trials and adjustments, the weight values for each category are determined as shown in Table IX.

Calculate the prior probability of each category according to (4). The results are shown in Table X. Add the weight values to the 400-dimensional feature embeddings obtained by the S-T model according to the (7), and then use them as the input embeddings of the cluster algorithm. The k-means cluster algorithm is used for clustering, and the calculation result of (3) is used as the posterior probability. So, the category of the movie can be judged by the obtained posterior probability.

After using the S-T method to express the word embeddings of the movie comment text, the obtained word embeddings are used as the input embeddings of the chasing clustering algorithm. The final movie clustering result is shown in Fig. 10. It can be seen that the model makes a good distinction between different types of movies, that is, our initial goal of clustering according to movie types is completed.

TABLE VIII. THE CHASING CENTER POINT OF THE CHASING CLUSTERING MODEL

Category	***Love***	***Comedy***	***Fantasy***	***Cartoon***	***Swordsman***
Center chasing point	Love	Amusemen t	Special effect	Innocence	Fight

TABLE IX. WEIGHT VALUES DISTRIBUTED BY CHASING-CLUSTERING MODEL

Category	***Love***	***Comedy***	***Fantasy***	***Cartoon***	***Swordsman***
Weight value ω	2	4	5	3	6

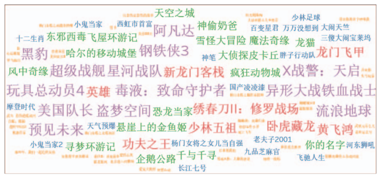


Figure 10. K-means clustering results of 100 movies.

TABLE X. THE PRIOR PROBABILITY OF CHASING-CLUSTERING ALGORITHM

Category	***Center chasing point***	***Number of documents with chasing center point***	***prior probability***
Love	Love	26	0.2476
Comedy	Amusement	19	0.1810
Fantasy	Special effect	17	0.1619
Cartoon	Innocence	22	0.2095
Swordsman	fight	10	0.0952

On the basis of using S-T method as the word embedding representation method, we have used k-means cluster algorithm and the chasing clustering algorithm proposed in this section to test the movie comment data experimentally, and used P , R and F_1 to evaluate the clustering effects. The results are shown in Table XI, Table XII, and Table XIII respectively.

As shown in Table XI, for the prediction results, the accuracy values of the swordsman in both models are higher. We believe that the main reason is the low dependency of the feature items of swordsman movies, which makes them very distinguishable. And in the chasing clustering model, the accuracy can reach 100%. The reason is that we further distribute a higher weight value to the center point fight of the swordsman class, and the number of movies whose center chasing point is fight is smaller at the same time. So, the predicted clustering results reach an accuracy of 100%.

As can be seen from Table XII, for the original sample, the recall rate of the love category has been significantly improved. We believe the main reason is that the chasing center point love also appears in movies of the remaining categories. Thus, the number of clustering results exceeds the number of original samples, and the original samples are successfully clustered into one category, so that the recall rate of the chasing cluster model reaches 100%.

Consider the accuracy and the recall rate, and use the significance evaluation index value F_1 to compare the two model algorithms. The results are shown in Fig. 11.

TABLE XI. ACCURACY OF S-T+K-MEANS MODEL AND CHASING CLUSTERING MODEL

Category	***S-T+k-means model***	***S-T+chasing clustering model***
Love	78.68%	48.61%
Comedy	41.95%	85.00%
Fantasy	73.12%	92.31%
Cartoon	61.13%	94.12%
Swordsman	90.00%	100%

TABLE XII. F_1 OF S-T+K-MEANS MODEL AND CHASING CLUSTERING MODEL

Category	***S-T+k-means model***	***S-T+chasing clustering model***
Love	55.00%	100.00%
Comedy	65.00%	85.00%
Fantasy	95.00%	85.00%
Cartoon	55.00%	80.00%
Swordsman	45.00%	45.00%

TABLE XIII. THE PRIOR PROBABILITY OF CHASING-CLUSTERING ALGORITHM

Category	***S-T+k-means model***	***S-T+chasing clustering model***
Love	64.70%	65.70%
Comedy	51.12%	85.00%
Fantasy	82.60%	88.51%
Cartoon	57.96%	86.49%
Swordsman	60.00%	62.07%

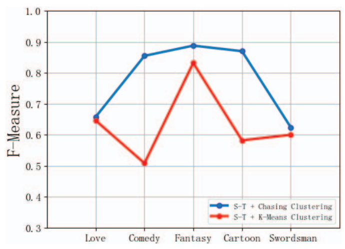


Figure 11. Comparison of the significance evaluation indexes F_1) between the S-T+k-means model and the S-T+chasing clustering model.

B. *Clustering of e-commerce platform comments targeting the quality of products*

The quality of products on e-commerce platforms is uneven, and the problem that the product entity does not match the description occurs from time to time. Therefore, the comments below products are important for consumers to differentiate. We will use the chasing clustering algorithm to cluster the quality of products through the comment data of different products.

The experimental data is product comments obtained from 200 commodities in Taobao website, and five chasing center points are set: "especially good", "good", "general", "bad", "too bad". The final clustering results are shown in Fig. 12(a).

It can be seen that the clustering results are basically consistent with the sales volume. Since the products in our data set are all women's dress, it can be considered that the sales volume and quality of the products are positively correlated. This means that through user comment data, our dividing results on product quality are relatively accurate.

In addition, we use the k-means clustering algorithm to cluster the same data set, and the results are shown in Fig. 12(b). It can be seen that the clustering result is not consistent with the quality of the product, and the result is more inclined to cluster different types of products, that is, the data samples are divided according to the categories of pants, skirts, short sleeves, coats, etc., and it is inconsistent with our original goal, which also reflects the effectiveness of the chasing clustering algorithm proposed in this paper.

(a) (b)

Figure 12. Commodity quality(a) and category(b) clustering results based on the chasing clustering algorithm.

V. Conclusion

Based on the existing word embedding representation method, this paper refers to the data characteristics of comment-type natural language, proposes an S-T method suitable for comment-type natural language word embedding representation, and applies the method to the movie clustering problem. It can be seen that the word embeddings generated by the S-T method improve the F_1-Score of the original cluster algorithm from 43.6% to 63.28%. So, the input embeddings constructed by the S-T method have a stronger saliency and representativeness for the comment-like natural language processing problem.

In addition, because the cluster algorithm is an algorithm that divides the samples according to the similarity of data, it is easy for the dividing results to be inconsistent with the target in the high-dimensional information space in the modern Internet field. In response to this, we propose a Bayesian-based chasing clustering algorithm. Limited by the center chasing points, it changes the conditional probability of the characteristic terms in the Bayes formula. The F_1-Score is raised from 63.28% to 77.55% by the chasing clustering algorithm, and the significant effect between the clustering results obtained is more obvious.

Finally, we applied the S-T method and chasing clustering algorithm proposed in this paper to the clustering problem of community comments targeting movie categories and the clustering problem of comments in e-commerce platforms targeting product quality, and obtained relatively satisfactory results.

References

[1] LIU Kan,YUAN Yunying. Short Texts Feature Extraction and Clustering Based on Auto-Encoder[J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2015, 51(2): 282-288.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Levy O , Goldberg Y . Neural word embedding as implicit matrix factorization[J]. Advances in neural information processing systems, 2014, 3:2177-2185.Sebastiani F. Machine Learning in Automated Text Categorization[J]. ACM Computing Surveys, 2002, 34(1):1-47.

[3] Siwei L , Kang L , Shizhu H , et al. How to Generate a Good Word Embedding[J]. IEEE Intelligent Systems, 2016, 31(6):5-14.

[4] Nyberg K, Raiko T, Tiinanen T, et al. Document Classification Utilising Ontologies and Relations between Documents[M]. Proceedings of the Eighth Workshop on Mining and Learning with Graphs. New York, NY, USA. 2010: 86‑93.

[5] Zhang D, Xu H, Su Z, et al. Chinese Comments Sentiment Classification Based on Word2vec and SVMperf[J]. Expert Systems with Applications, 2015, 42(4):1857-1863.

[6] Sebastiani F. Machine Learning in Automated Text Categorization[J]. ACM Computing Surveys, 2002, 34(1):1-47.

[7] Zhang H, Sheng S. Learning Weighted Naive Bayes with Accurate Ranking[C]//IEEE International Conference on Data Mining. 2004:567-570.

[8] Hall M. A Decision Tree-Based Attribute Weighting Filter for Naive Bayes[J]. Knowledge-Based Systems, 2007, 20(2):120-126.

[9] Katkar V D, Kulkarni S V. A Novel Parallel ImplemEntation of Naive Bayesian Classifier for Big Data[C]//International Conference on Green Computing, Communication and Conservation of Energy(ICGCE). Chennai, 2013:847-852.

[10] Jiang Q, Wang W, Han X, et al. Deep Feature Weighting in Mai-ve Bayes for Chinese Text Classification[C]//International Conference on Cloud Computing and Intelligence Systems(CCIS). Beijing, 2016:160-164.

[11] An S , Schorfheide F . Bayesian Analysis of DSGE Models[J]. Ssrn Electronic Journal, 2006.

[12] Gershman S, Blei D. A Tutorial on Bayesian Nonparametric Models[J]. Journal of Mathematical Psychology, 2011, 56(1):1-12