A Fine-Grained Sentiment Analysis Model for Product Reviews Based on Word2Vec and CNN (Postprint)

Authors: Cai Qingping, Ma Haiqun

Date: 2023-04-01T00:00:00+00:00

Abstract

[Purpose/Significance] Develop a fine-grained sentiment analysis model for product reviews based on Word2Vec and CNN. [Method/Process] First, Word2Vec is utilized to construct a product feature lexicon and a noise word list from product reviews. Second, the noise word list is leveraged to extract product review feature words. Then, CNN is employed for fine-grained sentiment classification at the product feature level. Finally, product review clustering based on product features is implemented. [Results/Conclusion] The model is trained and tested by crawling Huawei smartphone reviews from Jingdong Mall. The results demonstrate that the model can effectively achieve fine-grained sentiment analysis of product reviews and can effectively discover user attention and satisfaction regarding product features.

Full Text

A Fine-grained Sentiment Analysis Model for Product Reviews Based on Word2Vec and CNN

Cai Qingping[1], Ma Haiqun[2]
[1]School of Information Management, Heilongjiang University, Harbin 150080
[2]Research Center for Information Resource Management, Heilongjiang University, Harbin 150080

Abstract: [Purpose/Significance] This study constructs a fine-grained sentiment analysis model for product reviews based on Word2Vec and CNN. [Method/Process] First, Word2Vec is used to construct product feature word lists and noise word lists from product reviews. Second, the noise word list is utilized to extract product review feature words. Then, CNN is employed to perform fine-grained sentiment classification at the product feature level. Finally, product review clustering is implemented based on product features.

[Result/Conclusion] The model was trained and tested using Huawei phone reviews crawled from JD.com. Results demonstrate that the model can effectively achieve fine-grained sentiment analysis of product reviews and successfully identify user attention and satisfaction levels regarding product features.

Keywords: sentiment analysis; product reviews; CNN; Word2Vec; fine-grained

With the widespread application of internet technology in daily life, people increasingly shop online and publish product reviews on e-commerce platforms, typically disseminated as text. Through sentiment analysis of these product reviews, enterprises can grasp user satisfaction levels, identify core product strengths and weaknesses, and adjust product strategies accordingly, representing both new opportunities and challenges for businesses [?].

1 Related Research and Tool Selection

1.1 Sentiment Analysis

Sentiment analysis is the process of identifying emotional tendencies expressed by users from review text data. It can be categorized into coarse-grained and fine-grained sentiment analysis. Coarse-grained sentiment analysis evaluates overall sentiment polarity at the text level, providing a positive or negative assessment for an entire review. D. T. Vo et al. [?] incorporated emoji features to automatically construct sentiment lexicons for Twitter text analysis. D. Y. Tang et al. [?] expanded domain-specific sentiment words through seed words for user review classification. L. Zheng et al. [?] utilized document frequency for feature subset selection combined with support vector machines for Chinese product review sentiment analysis.

Fine-grained sentiment analysis, by contrast, examines sentiment polarity at the text segment or feature level, providing separate positive or negative evaluations for each feature. I. Titov et al. [?] constructed a multi-aspect sentiment model (MAS) based on multi-granularity LDA. Sun Yan et al. [?] built an unsupervised hybrid sentiment model. Y. Kim [?] applied convolutional neural networks to multiple sentiment analysis datasets, achieving excellent results. Li Jie et al. [?] used CNN for short text review sentiment analysis to provide decision support for product optimization.

Although product reviews are brief, they involve multiple product features, and user attitudes toward each feature may differ. Therefore, product review sentiment analysis is well-suited for fine-grained sentiment analysis. This paper employs a convolutional neural network model to implement fine-grained sentiment analysis of product reviews.

1.2 Product Review Feature Word Extraction

Product review feature word extraction refers to identifying words related to product features from review texts, typically including product functions and performance. Many scholars have conducted related research on this topic. Yu Chuanming et al. [?] extracted product feature words from customer review texts using support vector machines. Xu Jianmin et al. [?] implemented feature word extraction from texts using the TF-IDF method based on an ontology library. Xia Tian [?] integrated word vector technology into the TextRank method to improve feature word extraction effectiveness.

With the widespread application of deep learning in sentiment analysis, shallow neural network models based on probability have emerged [?]. Word2Vec represents an outstanding example of such models, with increasing numbers of scholars researching its use for text feature word extraction. Li Yuepeng et al. [?] proposed a feature word extraction algorithm using Word2Vec technology combined with k-means clustering. Zhou Shunxian et al. [?] used Word2Vec to construct a word vector clustering centroid frequency model for text feature word extraction. For fine-grained sentiment analysis of product reviews, feature word extraction is particularly important. This paper attempts to use Word2Vec for product review feature word extraction to improve extraction effectiveness.

1.3 CNN (Convolutional Neural Network)

This paper selects CNN (convolutional neural network) [?, ?] as the sentiment classifier for product reviews. Figure 1 [Figure 1: see original paper] shows the CNN model structure for product review text classification, which consists of an input layer, convolutional layer, pooling layer, and fully connected layer.

Figure 1. CNN Model Structure

(1) **Input Layer:** This layer comprises a matrix formed by word vectors corresponding to words in the product review text.

(2) **Convolutional Layer:** This layer' s primary function is to perform convolution operations on the input word vector matrix using convolutional kernels to obtain deeper-level text features.

(3) **Pooling Layer:** This layer performs feature selection and information filtering on text vectors extracted by the convolutional layer, retaining main features while reducing parameters and computation in the next layer to prevent overfitting.

(4) **Fully Connected Layer:** This layer connects the feature values corresponding to the most significant text features from m product reviews obtained from the pooling layer to produce a fixed-length feature vector representing product review features. This feature vector is then input into the final softmax regression classifier for global feature analysis to complete sentiment classification of product feature reviews.

1.4 Word2Vec

Word2Vec [?, ?, ?, ?] is a word semantic computation technique proposed by Google in 2013. Through Word2Vec training, text processing can be simplified to vector operations in K-dimensional vector space, where similarity in vector space can represent semantic similarity. Word2Vec provides two classic language models for training: CBOW and Skip-gram. For these models, Word2Vec offers two frameworks based on Hierarchical Softmax and Negative Sampling. This paper adopts the Skip-gram model based on Hierarchical Softmax.

Hierarchical Softmax uses a Huffman tree structure to represent the output layer words, where the W words in the output layer exist as leaf nodes, with each node representing the relative probability of its child nodes. In a Huffman tree, there is always an optimal path from the root node to each leaf node w.

The Skip-gram model consists of a three-layer network: input layer, projection layer, and output layer, as shown in Figure 2 [Figure 2: see original paper]. The training objective of the Skip-gram model is to find word representations that help predict surrounding words in sentences or documents.

Figure 2. Skip-gram Model Structure

2 Fine-grained Sentiment Analysis Model for Product Reviews Based on Word2Vec and CNN

This paper designs a fine-grained sentiment analysis model for product reviews based on Word2Vec and CNN, comprising five modules: product review preprocessing, product feature vocabulary and noise vocabulary construction, product review feature word extraction, product review sentiment classification, and product review clustering based on product features, as detailed in Figure 3 [Figure 3: see original paper].

Figure 3. Fine-grained Sentiment Analysis Model for Product Reviews Based on Word2Vec and CNN

2.1 Product Review Preprocessing

Product review datasets often contain meaningless reviews that include no feature words, such as those containing only words like "good," "bad review," "not bad," or "like." Filtering these useless reviews helps reduce noise interference and improves text sentiment classification and feature word extraction accuracy.

Unlike English, which uses spaces as word delimiters, Chinese lacks clear boundaries between words. Therefore, Chinese product reviews must first undergo word segmentation and part-of-speech tagging, using words as the basic components of product reviews.

After segmentation and part-of-speech tagging, stop word lists are used to remove common stop words from product reviews, including words like "suddenly," "immediately," "not only," "but also," and "we" that are unrelated to product features and sentiment classification.

2.2 Construction of Product Feature Vocabulary and Noise Vocabulary

2.2.1 Product Feature Vocabulary Construction Product features refer to product attributes or functions. Taking mobile phones as an example, screen, appearance, memory, and camera all constitute product features. The product feature vocabulary serves as the basis for constructing noise vocabulary and clustering product reviews based on product features. Product feature vocabulary construction involves three steps:

(1) **Initial Product Feature Word Extraction:** Since most product attribute words are nouns and noun phrases [?], and high-frequency nouns and noun phrases often represent genuine feature words, this paper conducts frequency statistics on nouns and noun phrases in preprocessed product reviews, manually extracts the top m high-frequency nouns and noun phrases related to product features, classifies these feature words by product feature category, and forms the initial product feature vocabulary Ts.

(2) **Word Vector Training:** Inter-word similarity can be conveniently measured by calculating cosine distance between word vectors, so word vectors represent deep semantic relationships between words in the corpus. Word2Vec uses the vocabulary constructed from text data as training data to learn high-dimensional vector representations of words, mapping words into a finite-dimensional high-dimensional space. Training Word2Vec on the segmented product review dataset yields a word vector model and vector representations of each word with specified dimensions. Word vectors can be used for both product feature extraction and as input for convolutional neural networks.

(3) **Product Feature Vocabulary Generation:** Multiple words may describe the same product feature. For example, words describing mobile phone appearance include "appearance," "design," "style," "aesthetics," and "model." Therefore, to obtain comprehensive feature words, feature word clustering is necessary. This paper uses the word vector model obtained from Word2Vec training to calculate similarity between different words in the product review dataset and each feature word in the initial product feature vocabulary Ts. For each product feature word, the top n words with high similarity related to product features are selected to expand the initial product feature vocabulary Ts, thereby completing the generation of product feature vocabulary T.

2.2.2 Noise Vocabulary Construction Since feature words extracted from product reviews may not all be genuine feature words, this paper considers those unrelated to product features as noise words that interfere with product review clustering based on product features. Therefore, this paper constructs a noise vocabulary to filter feature word sequences extracted from product review texts, removing noise words to correctly extract product review feature words. The noise vocabulary construction process is as follows:

Input: Product feature vocabulary T and product review set D' s word vector model after Word2Vec training.

Step 1: Read any feature word wi from T.
Step 2: Calculate the top n words with high similarity to wi.
Step 3: Identify words unrelated to product features from these n words and add them to noise vocabulary Z.
Step 4: Repeat Steps 1, 2, and 3 until every product feature word in T has been processed.

Output: Noise vocabulary Z.

2.3 Product Review Feature Word Extraction

The product review feature word extraction method in this paper is implemented based on the Skip-gram model with Hierarchical Softmax. The process is as follows:

Input: Preprocessed product review set D = {S1, S2, ···, Sm}, where Sk = {w1, w2, ···, wi, ···, wnk} for k = 1, 2, ···, m.

Step 1: Read any product review Sk from D. For each word wi in Sk, calculate p(wj|wi) where j = 1, 2, ···, nk.
Step 2: For each word wi in Sk, calculate p(Sk|wi) = p(w1, w2, ···, wnk|wi) = p(wj|wi).
Step 3: Sort all words in Sk by p(Sk|wi) in descending order and select the top n words as Sk' s feature words.
Step 4: Repeat Steps 1, 2, and 3 until all product reviews in D have undergone feature word extraction, forming product review feature word list Dkey.
Step 5: Use noise vocabulary Z to filter the product review feature word list Dkey corresponding to product review set D, finally obtaining the denoised product review feature word list Dc.

Output: Product review feature word list corresponding to product review set D.

2.4 Product Review Sentiment Classification

2.4.1 Product Review Text Segmentation Each product review may contain multiple product features, each with potentially different sentiment labels. For example, the review "Screen is large enough, smoothness is good, signal is good, but battery doesn't last" contains features "screen," "performance," "signal," and "battery," with positive sentiment labels for "screen, performance, signal" and a negative label for "battery."

Product review texts often contain multiple delimiters, including periods (.), commas (,), exclamation marks (!), hash symbols (#), asterisks (*), and common emoticons. Users' short sentences expressing opinions on specific product features typically appear within one delimiter. Therefore, this paper adopts a delimiter-based segmentation strategy (BSP) [?] to split each product review into multiple short sentences, filter out sentences without product feature words, and merge multiple sentences describing the same product feature into a new short sentence, ensuring each final short sentence evaluates only one product feature. After segmentation, product review set D forms a new product review set Dw.

2.4.2 CNN Model Training and Testing A portion of product review data is extracted from product review set Dw as the training set for CNN model sentiment classification training. Training set samples have been manually annotated with sentiment labels. The CNN model calculates sentiment labels for training samples and compares them with existing labels, using computational errors to continuously adjust model parameters. Another portion of product review data is extracted from Dw as the test set to evaluate the CNN model's sentiment classification effectiveness, i.e., using the CNN model to calculate sentiment labels for test set reviews and comparing them with manual annotations.

2.4.3 Product Review Sentiment Classification After CNN model training and testing, unlabeled review texts in product review set Dw can be sentiment classified, with sentiment labels added to each product review.

2.5 Product Review Clustering Based on Product Features

After sentiment classification by the CNN model, each product review in Dw contains corresponding sentiment labels, and the feature words within also carry sentiment labels. To determine which product features each review positively or negatively evaluates, this paper first extracts feature words from each product review in Dw, then clusters product reviews based on product features to analyze and evaluate user attention and satisfaction regarding product features. The clustering process is as follows:

Input: Product review set Dw and product feature vocabulary T = {T1, T2, ···, Tq}, where Tj represents the feature word list for the j-th product feature.

Step 1: Extract feature words from each product review in Dw to generate product review feature word list Dsplit = {S1, S2, ···, Sp}.
Step 2: Read any line Si from Dsplit, where Si = {w1, w2, ···, wni} (i = 1, 2, ···, p).
Step 3: Read any feature word wi from Si.
Step 4: Read any Tj from T, where Tj = {t1, t2, ···, tnj} (j = 1, 2, ···, q).
Step 5: Read any feature word to from Tj.
Step 6: Calculate similarity Simo between wi and to.
Step 7: Repeat Steps 5 and 6, recording the maximum value maxij =

max({Sim1, Sim2, ···, Simnj}).
Step 8: Repeat Steps 4 through 7 to calculate max({maxi1, maxi2, ···, maxiq}), determining the feature category i to which feature word wi belongs.
Step 9: Repeat Steps 3 through 8 until all words in Si have been processed, determining the product feature category to which review text Si belongs.
Step 10: Repeat Steps 2 through 9 until all lines in Dsplit have been processed.

Output: Product feature categories for each product review.

3 Experiments and Results Analysis

3.1 Experimental Environment

All experiments in this paper were completed in the environment shown in Table 1 .

3.2 Product Review Preprocessing

This paper collected 10,223 Huawei phone product reviews from JD.com using a Python web crawler. These reviews were used for model application and results analysis. First, meaningless product reviews were cleaned, resulting in a filtered product review set of 9,230 reviews.

Second, the Python jieba segmentation package was used for word segmentation and part-of-speech tagging of review texts. Finally, based on the Harbin Institute of Technology stop word list, frequency statistics were conducted on words in the segmented product review set, and high-frequency words unrelated to product features and sentiment words were selected to construct a stop word list suitable for phone reviews, which was then used to remove stop words from the segmented product review set.

3.3 Construction of Product Feature Vocabulary and Noise Vocabulary

3.3.1 Word Vector Training The segmented product review set was used as training corpus for word vectors. Python' s gensim package was used to train word vectors and construct a word vector model. Table 2 shows Word2Vec parameter settings, and Table 3 displays word vectors for some words in the product review set.

3.3.2 Generation of Product Feature Vocabulary and Noise Vocabulary First, frequency statistics were conducted on nouns and noun phrases in the preprocessed product review set, and the top 200 high-frequency nouns and noun phrases related to phone features were extracted as initial product feature words. These feature words were classified by product feature category to form the initial product feature vocabulary, shown in Table 4 .

Second, Python' s gensim package Word2Vec function most_{similar}() was used to calculate similarity between nouns/noun phrases in the product review set and initial product feature words. From the top 50 words with high similarity to product feature words, words truly related to product features were selected to expand the initial product feature vocabulary. Figure 4 [Figure 4: see original paper] shows the calculation process for the top 50 words similar to feature word "network," with results in Table 5 . Words related to "network" from Table 5 were selected to expand the initial product feature vocabulary, generating the product feature vocabulary shown in Table 6 .

Finally, the noise vocabulary was constructed based on the product feature vocabulary, as shown in Table 7 .

3.4 Product Review Feature Word Extraction

(1) Using the trained word vector model, product review feature words contained in each product review were extracted to form a product review feature word list corresponding to the product review set, as shown in Table 8 .

(2) The noise vocabulary was used to filter noise words from the product review feature word list, generating a denoised product review feature word list, as shown in Table 9 .

For comparative analysis with existing feature word extraction methods, this paper adopts precision (Pextract), recall (Rextract), and F1-score (F1extract) as evaluation metrics for feature word extraction effectiveness. Pextract, Rextract, and F1extract are calculated as shown in formula (2) [**?**, **?**], where ci represents the feature word set extracted by the feature extraction method for the i-th product review, di represents the feature word set attached to the i-th product review, and M represents the number of product reviews in the dataset.

Five hundred product reviews were selected from the product review set and processed using TF-IDF, TextRank, and the method proposed in this paper. All three methods set the number of extracted feature words to 2-10, and the three evaluation metrics were calculated and compared. Results are shown in Figures 5 [Figure 5: see original paper], 6 [Figure 6: see original paper], and 7 [Figure 7: see original paper].

Figures 5, 6, and 7 show that when using the proposed feature word extraction method with different numbers of extracted feature words, both precision (Pextract) and F1-score (F1extract) outperform TF-IDF and TextRank methods, while recall (Rextract) is also comparable to TF-IDF and TextRank when the number of feature words is 8-10. Therefore, for short texts like product reviews, the proposed noise vocabulary filtering-based feature extraction method demonstrates better performance.

3.5 Product Review Sentiment Classification

Before CNN model training, parameters must be determined. The CNN model primarily has the following parameters: (1) convolutional kernel size, which determines the region size for convolution on input word vectors. Since words are the smallest linguistic units and only contextual effects need consideration, convolutional kernel width remains maximally unchanged while only height varies; (2) number of convolutional kernels; (3) dropout ratio, i.e., the proportion of data set to zero; (4) batch size for stochastic gradient descent. This paper uses grid search [?] to determine these parameters, while the learning rate for stochastic gradient descent is adaptively adjusted using the Adadelta update rule. Final parameter values are shown in Table 10 .

Each product review in the product review set was first segmented, with results shown in Table 11 . The segmented product review set was then preprocessed and divided into training, validation, and test sets in a 6:2:2 ratio. The training set fits the model by training the classifier with set parameters; the validation set uses the trained model to predict validation data for parameter tuning to select the best-performing model; the test set uses the trained CNN model to sentiment classify unlabeled product reviews.

Evaluation metrics for sentiment classification experiments include traditional information retrieval metrics: precision (Precision), recall (Recall), and F1-score (F1-score) [?]. SVM and NB (Naive Bayes) were set as control groups for comparison with CNN, calculating Precision, Recall, and F1-score to compare classifier performance. CNN sentiment classification results and comparison with control experiments are shown in Table 12 . Table 12 shows that CNN achieves better performance in Precision, Recall, and F1-score for product review sentiment classification.

3.6 Comment Clustering Based on Product Features

After segmentation, the product review set generates a product review short sentence set. These short sentences, after sentiment classification by the CNN model, carry corresponding sentiment labels for the product features described in each sentence. Using Python’ s gensim package Word2Vec similarity() function, similarity is calculated between feature words in each short sentence and feature words for each product feature category in Table 6. The feature category with maximum similarity is determined as the belonging category for the current short sentence, thereby clustering product reviews based on product features for statistical analysis. Results are shown in Figures 8 [Figure 8: see original paper] and 9 [Figure 9: see original paper].

3.7 Analysis of Experimental Results

Figures 8 and 9 show the distribution and sentiment tendencies of user reviews regarding Huawei phone features. Figure 8 indicates that users primarily focus on screen, performance, appearance, and functionality. Figure 9 shows that users are highly satisfied with phone performance and appearance, but have many complaints about network, call quality, and accessories, with mixed reviews for screen, camera, and battery.

In terms of performance, Huawei' s self-developed Kirin series processors approach Qualcomm Snapdragon series performance, with Huawei phones offering relatively higher configurations at the same price point, resulting in excellent performance. In appearance, Huawei' s Honor series adopts colorful glass body designs that are fashionable and attractive, aligning with young people' s aesthetic preferences. Designers should maintain these design solutions.

Huawei' s Leica camera meets user needs, but issues like slow photo response and suboptimal night imaging require improvement. Users also express concerns about screen calibration, fingerprint recognition, low sensitivity, and fragility. In functionality, features like double-tap lock screen, NFC, and fingerprint payment are missing. Regarding battery, users generally report that although capacity is 4000mAh, performance seems more like 3200mAh, with poor battery life and serious heating during charging. Designers should address these shortcomings while maintaining strengths.

For features receiving many negative evaluations like network, call quality, and accessories, designers should quickly identify causes and make improvements. For example, users report weak WiFi signal reception, frequent 4G network lag, and disconnection during gaming. Since Huawei' s main business is communications, user expectations for network features are high. For call quality, users frequently report noise, unclear sound quality, and low volume. Accessories like buttons, earphones, earpieces, and protective films are easily damaged. Improvements in these areas can further enhance phone competitiveness.

In today' s internet era, online platforms contain vast amounts of product reviews, making fine-grained sentiment analysis crucial for both users and enterprises. On one hand, users can understand others' experiences and evaluations of product features to make better purchase decisions. On the other hand, product reputation is vital for enterprises—positive reputation aids product promotion and sales, while negative reputation damages corporate image. Therefore, enterprises can dynamically monitor changes in user reputation through fine-grained sentiment analysis of product reviews, enabling timely warnings and responses when negative reputation emerges to reduce corporate impact.

This paper studies online product review sentiment analysis and proposes a fine-grained sentiment analysis model based on Word2Vec and CNN. The model implements product review feature word extraction, feature-level sentiment classification, and clustering analysis based on product features. It can effectively capture sentiment information about product features in reviews, efficiently mine user attention and satisfaction regarding product features, and serve enterprise management and decision-making.

The innovation lies in proposing a Word2Vec-based, noise vocabulary-filtering feature word extraction method tailored to short text characteristics of product reviews. This method fully considers contextual semantic information and further improves feature word extraction effectiveness. The model performs sentiment classification at the product feature level and clusters sentiment-labeled reviews by product features, facilitating analysis of feature-specific sentiment tendencies and providing an effective approach for fine-grained sentiment analysis of product reviews.

This paper adopts binary sentiment classification (positive/negative) for product reviews. Future research could explore multi-class sentiment classification to enable judgment of sentiment intensity.

References

[1] Wang Renwu, Song Jiayi, Chen Chuanbao. Research on the application of Word2Vec-based sentiment analysis in brand cognition [J]. Library and Information Service, 2017, 61(22): 6-12.
[2] VO D T, ZHANG Y. Don' t count, predict! an automatic approach to learning sentiment lexicons for short text [C] // YANNICK V, HAIZ, YUSUKE M. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2016: 219-224.
[3] TANG D Y, QIN B, ZHOU L J, et al. Domain-specific sentiment word extraction by seed expansion and pattern generation [J/OL]. [2019-11-26]. http://arxiv.org/pdf/1309.6722.pdf.
[4] ZHENG L, WANG H, GAO S. Sentimental feature selection for sentiment analysis of Chinese online reviews [J]. International journal of machine learning and cybernetics, 2018, 9(1): 75-84.
[5] TITOV I, MCDONALD R. Modeling online reviews with multigrain topic models [C] // LIZ. Proceeding of the 17th international conference on World Wide Web. New York: ACM, 2008: 111-120.
[6] Sun Yan, Zhou Xueguang, Fu Wei. Unsupervised text sentiment analysis based on topic-sentiment mixture model [J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2013, 49(1): 102-108.
[7] KIM Y. Convolutional neural networks for sentence classification [C] // YUVAL M. Proceedings of the 2014 conference on empirical methods in natural language processing. Doha: ACL, 2014: 1746-1751.
[8] Li Jie, Li Huan. Research on product feature extraction and sentiment classification of short text reviews based on deep learning [J]. Information Studies: Theory & Application, 2018(2): 143-148.
[9] Yu Chuanming, Chen Lei, Zhang Xiaoqing. Research on product attribute recognition based on support vector machine [J]. Journal of the China Society for Scientific and Technical Information, 2010, 29(6): 1038-1044.
[10] Xu Jianmin, Wang Jinhua, Ma Weiyu. Improved TF-IDF feature word extraction method using ontology correlation [J]. Information Science, 2011, 29(2): 279-283.
[11] Xia Tian. Keyword extraction using word vector clustering weighted TextRank [J]. Data Analysis and Knowledge Discovery, 2017(2): 28-33.

[12] Zhang Liu, Wang Xinwei, Huang Bo, et al. Research and experiment on sentiment classification model for Weibo comments based on character-word vector and multi-scale convolutional neural network [J]. Library and Information Service, 2019, 63(18): 99-108.
[13] Li Yuepeng, Jin Cui, Ji Junchuan. Keyword extraction algorithm based on Word2Vec [J]. e-Science Technology & Application, 2015, 6(4): 54-59.
[14] Zhou Shunxian, Jiang Li, Lin Shuangqiao, et al. Text feature representation method based on Word2Vector [J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2018, 30(2): 272-279.
[15] Jin Zhigang, Hu Bohong, Zhang Rui. Multi-dimensional feature microblog sentiment analysis based on deep learning [J]. Journal of Central South University (Science and Technology), 2018(5): 1135-1140.
[16] Lai Wenhui, Qiao Yupeng. Spam SMS recognition method based on word vector and convolutional neural network [J]. Journal of Computer Applications, 2018, 38(9): 2469-2476.
[17] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [J/OL]. [2019-11-26]. http://arxiv.org/pdf/1301.3781v3.pdf.
[18] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C] // BURGES C J C, BOTTOU L, WELLING M. Proceedings of the 26th international conference on neural information processing systems. Lake Tahoe: NIPS, 2013: 3111-3119.
[19] Ning Jianfei, Liu Jiangzhen. Research on keyword extraction integrating Word2Vec and TextRank [J]. Data Analysis and Knowledge Discovery, 2016(6): 20-27.
[20] Zhang Jian, Qu Dan, Li Zhen. Recurrent neural network language model based on word vector features [J]. Pattern Recognition and Artificial Intelligence, 2015(4): 299-305.
[21] WU Y B, ZHANG Q, HUANG X J, et al. Phrase dependency parsing for opinion mining [C] // DAVID C. Proceedings of the 2009 conference on empirical methods in natural language processing. Singapore: ACL, 2009(3): 1533-1541.
[22] Li Zhiyu, Liang Xun, Zhou Xiaoping. Optimization algorithm for constructing review short text word vectors based on attribute topic segmentation [J]. Journal of Chinese Information Processing, 2016(9): 101-110.
[23] Liu Qifei, Shen Weiyu. Research on keyword extraction method for current political news based on Word2Vec and TextRank [J]. Information Research, 2018(6): 22-27.
[24] Xu Chen, Cao Hui, Zhao Xiao. Parameter selection method for speaker recognition based on SVM [J]. Computer Engineering, 2012(11): 175-177.

Author Contributions

Cai Qingping: Data collection and processing, experiments, initial draft writing and revision; Ma Haiqun: Topic selection and framework development, paper revision and review.