Article

Stacking-Based Ensemble Learning of Self-Media Data for Marketing Intention Detection

Yufeng Wang [1,†]**, Shuangrong Liu** [1]**, Songqian Li** [1]**, Jidong Duan** [1]**, Zhihao Hou** [1]**, Jia Yu** [1] **and Kun Ma** [1,2,*,†]

[1] School of Information Science and Engineering, University of Jinan, Jinan 250022, China
[2] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan 250022, China
[*] Correspondence: ise_mak@ujn.edu.cn; Tel.: +86-0531-89736863
[†] These authors contributed equally to this work.

check for updates

Abstract: Social network services for self-media, such as Weibo, Blog, and WeChat Public, constitute a powerful medium that allows users to publish posts every day. Due to insufficient information transparency, malicious marketing of the Internet from self-media posts imposes potential harm on society. Therefore, it is necessary to identify news with marketing intentions for life. We follow the idea of text classification to identify marketing intentions. Although there are some current methods to address intention detection, the challenge is how the feature extraction of text reflects semantic information and how to improve the time complexity and space complexity of the recognition model. To this end, this paper proposes a machine learning method to identify marketing intentions from large-scale We-Media data. First, the proposed Latent Semantic Analysis (LSI)-Word2vec model can reflect the semantic features. Second, the decision tree model is simplified by decision tree pruning to save computing resources and reduce the time complexity. Finally, this paper examines the effects of classifier associations and uses the optimal configuration to help people efficiently identify marketing intention. Finally, the detailed experimental evaluation on several metrics shows that our approaches are effective and efficient. The F1 value can be increased by about 5%, and the running time is increased by 20%, which prove that the newly-proposed method can effectively improve the accuracy of marketing news recognition.

Keywords: marketing intention; feature extraction; ensemble learning

1. Introduction

1.1. Background

New media are forms of media that are native to computers and mobile phones for redistribution. Some examples of new media are virtual worlds, social networking, Internet-oriented websites, and self-media platforms [1]. With the rapid development of information processing technology and network transmission technology, it has entered an unprecedented era of information explosion. In the absence of supervision of the authenticity of marketing news on the Internet, malicious marketing of media posts, including false or untrue news, poses potential harm for society, and the identification of marketing intentions becomes particularly important. It is a good idea to use news classification to identify news with marketing intentions. News classification is a technique for classifying news and automating the newly generated news. News classification belongs to a kind of supervised machine learning technology. The algorithm needs to train the model according to the news content that has been marked with the correct category and then use the model to classify the news of unknown

categories automatically. Against this background, this paper has proposed an efficient ensemble learning method to identify marketing intention, which is used to find the marketing news on the Internet and provide a reference value for identifying marketing news.

1.2. Challenges

The key to detecting marketing intention is to analyze the news text and make marketing classification of news text by characteristics. The important parts of text classification are feature representation and the classification algorithm, which have a huge influence on text classification. Feature representation is generally based on a bag-of-words model. In the process of transforming text into a high-dimensional vector space model, the contextual structure information of words is effectively expressed as a challenge. Another challenge is that the stacking strategy is complex for obtaining the classification algorithm model with high accuracy. Therefore, it is necessary to simplify the model to reduce the running time of the model.

1.3. Contributions

The contributions of our work are latent feature extraction and stacking-based classification.

- Optimization of Latent Semantic Analysis (LSI): The feature extraction model of LSI-Word2vec is proposed: combining the advantages of Word2vec [2] in document expression. The combined model not only can express the context features efficiently, but can express the semantic features of the words efficiently, and the text feature expression capability of the model is heightened.
- Simplification of stacking fusion: A set pruning method based on the decision tree classifier is proposed: using the model of the decision tree after pruning can reduce the computing resources of integrated learning and enhance the learning effect.

1.4. Organization

The rest of the paper is planned as follows: the reasons for the selection of text data feature methods and classifier models will be discussed in the relevant work in Section 2. The third part introduces the overall architecture process based on the stacking ensemble learning model, and we explain the steps of our method in detail. In the fourth part, various model experiments show that our method is efficient. The related work of this paper is summarized in the last section.

2. Related Work

2.1. Feature Extraction

Traditional text features are based on a word package model and the Term Frequency-Inverse Document Frequency (TF-IDF) statistical method. Word2vec is commonly used in various news reports, and it has been used for such aspects as keywords' extraction and Chinese lexicon extraction. Word2vec is able to marry relevant words in descending order so that people are capable of comprehending the meaning of the news [2]. Experiments were performed by drawing the TF-IDF characteristics from the news. The experiments showed that the features were appropriate for highlighting the helpful keywords within news, which can be beneficial for research findings on major diseases [3]. TATF-IDF is an algorithm for solving document frequency (TF-IDF) based on word frequency. This algorithm finds popular words according to the distribution of the attention and time of users. A method was proposed to produce brand nomenclatures and assorted nomenclatures, which were divided by the Chinese word segmentation algorithm. Based on the improved tf-idf algorithm, hot event problems could be found effectively [4]. The popular method of the characteristics of the processing is based on semantics. Multi-Co-Training (MCT) was put forward to advance the manifestation of document categorization. So as to add the diversity of feature sets for classification, a document using three document expression means was switched: (TF-IDF based on the bag-of-words strategy, theme

distribution on account of the Latent Dirichlet Allocation (LDA), and neural-network-based document embedding referred to as Document to Vector (Doc2Vec). MCT is robust to the variation of parameters, and compared with the benchmark method, it achieved good performance [5]. To investigate the effect of corpus specificity and size in word embeddings, two ways were studied for progressive elimination of documents: the elimination of random documents vs. the elimination of documents unrelated to a specific task. Results showed that Word2vec can take advantage of all the documents, obtaining its best performance when it was trained with the whole corpus. On the contrary, the specialization (removal of out-of-domain documents) of the training corpus, accompanied by a decrease of dimensionality, can increase LSI word-representation quality while speeding up the processing time [6]. The precise mechanisms of Word2vec can make full use of all the documents and obtain the best performance through the training of the whole corpus. LSI not only reduces the dimensionality, but also identifies latent dimensions based on singular values [7]. However, existing methods are weakened for expression and theme extracting. Therefore, we set up LSI-Word2vec for feature extraction and efficient theme extracting.

2.2. Text Classification

A denial framework for the naive Bayes classifier was proposed by using the posterior probability for determining the confidence score. The presented way was assessed on the WebKB datasets. This means of classification can achieve a filtering function, which is efficient [8]. A classification algorithm of metric data on account of Support Vector Machine (SVM) has been used. SVM and the Kalman filter are grouped to gain updated classification at every sampling stage, and experimental and quantitative results from the actual data proved the efficiency and consistency of the result compared with existing similar means [9]. An automatic classification system of text based on a genetic algorithm has been developed. Genetic algorithm classifiers generate pre-defined optimal classification methods, which have strong flexibility, including most of the characteristics of the training sets. The performance of genetic algorithm is better than the k-nearest neighbor classifier [10]. The text classification model is changed from initiation to the goal domain by a deep neural network learning transferable traits. The conventional domain orientation has been applied to fit the models of a simplex specific domain with enough tagged data to another single specific target domain with no marked data. Antagonistic training tactics have been put forward to ensure that private and joint features do not clash with each other in support of enhancing the nature of the source and target domains. It is proven that this way can be used for all kinds of scenarios in a dataset that is marked with respect consumptive purposes [11]. Comparing stacking with a single model, stacking has the ability of expression, which is independent and nonlinear. Stacking combines finite models and then estimates the nonparametric density, so the generalization error can be reduced. Therefore, in the Kaggle [12] competition, generally, high-scoring teams adopted stacking to achieve excellent results [13–15]. However, the stacking model structure is complex; it has long training time as the base model, and the model needs to be simplified. This paper proposes a method of using a decision tree as the base classifier, which is inherently able to mix static data.

The decision tree for pruning can simplify the stacking model and reduce the operation time [16]. A scalable end-to-end tree boosting system called XGBoost is described, which is used widely for data by scientists to achieve state-of-the-art results on many machine learning challenges [17]. LightGBM is appleied for human action identification. Compared with simplex SDAE algorithm or CNN (convolutional neural network), it has a good effect [18]; XGBoost uses a second-order gradient for node partitioning with higher precision. A characteristic discretization based on Extreme Gradient Boosting (XGBOOST) is put forward to settle trending questions on weibo with further quantity of retransmission and criticism at a certain time. Decentralizing the characteristic by the prediction way of the tree, The model can improve the operation effciency [19]. Gradient boosting decision tree (GBDT) has come forward in this research process for the discovery of Hand, foot, and mouth disease (HFMD). A lacking data input method on account of the GBDT model was put forward.

This model has made a good contribution to the identification of HFMD, and the model can be used for identifying severe HFMD better [20]. RF features can improve the classification ability of the algorithm. A multi-classifier system based on random forest and potential nearest neighbor method has been put forward. The classification manifestation of random forest is determined by the diversity of students and the intensity of weak learners. Compared with rotating forests and other basic random forests, the means of using this method can have lesser variance [21].

3. Approach

The architecture of marketing intention recognition is shown in Figure 1, which is mainly composed of four parts: text preprocessing, feature extraction, classifier, and output. First, the data are processed to remove stop words and form word segmentation, and then, LSI-Word2vec is used for feature extraction in the second stage. Finally, the classifier is used for recognition, and the detailed process will be described in the next subsections.
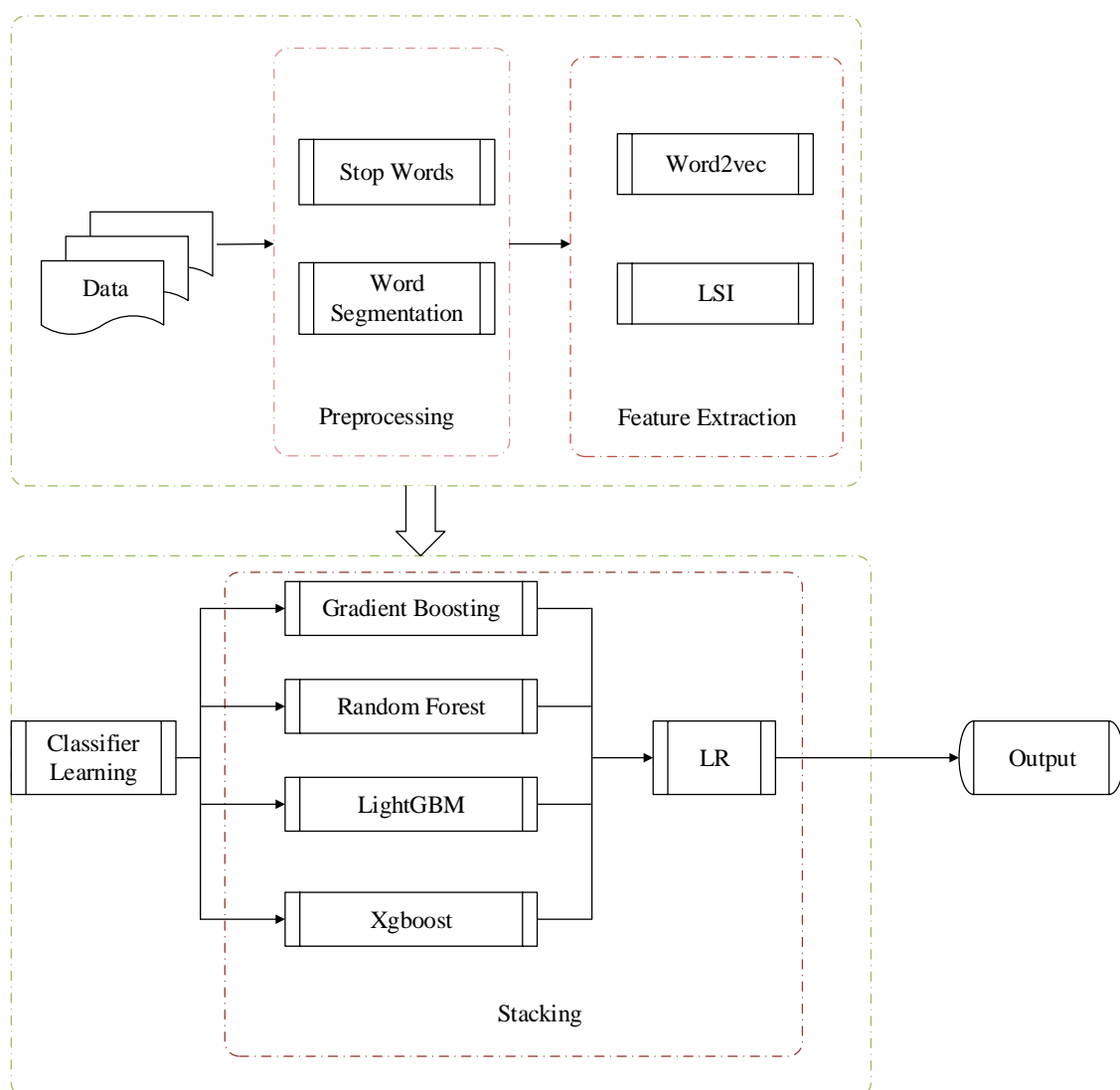

Figure 1. General architecture. LSI, Latent Semantic Analysis.

3.1. Preprocessing

First, the crawled HTML tags are removed to generate only texts [22]. Then, the scanning of words and figures is realized based on the Trie tree [23]. A directed acyclic graph is generated according to all possible word combinations in the sentence. Dynamic programming is adopted to find the

maximum probability path, so as to find the maximum segmentation and combination process based on word frequency. The Hidden Markov Model (HMM) [24] is used to solve the unlogged words. That is, four implicit states are used, which are represented as a single word, the beginning of a phrase, the middle of a phrase, and the end of a phrase. Various parameters of HMM could be obtained through the marked word segmentation training set. Then, the Viterbi algorithm is used to get word sequences. Finally, a preprocessing file is formed after removing stop words. The process is shown in Figure 2.
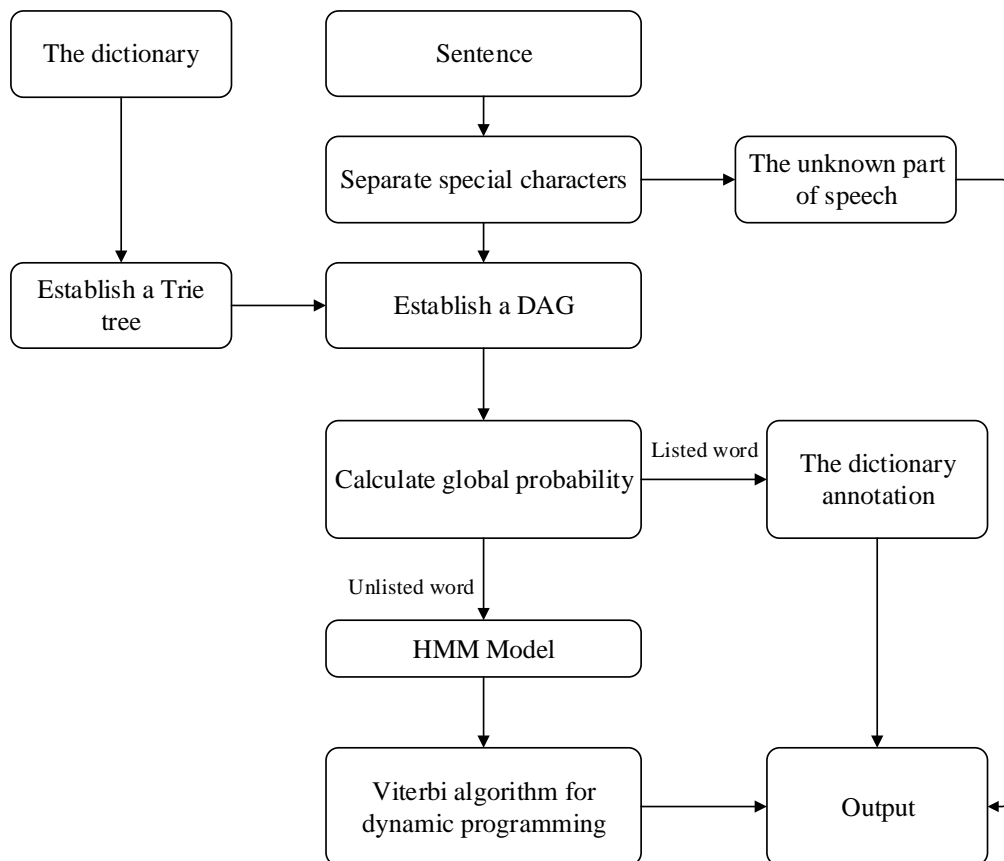

Figure 2. Word segmentation.

3.2. Feature Extraction

3.2.1. LSI-Word2vec

Word2vec uses two models, skip-gram and Continuous Bag-Of-Words(CBOW) model. The CBOW model uses the C(C = 2) words before and after $V(t)$ to predict the current word; the skip-gram model is just the opposite. It uses the word $V(t)$ to predict words before and after it. In this paper, if $V(t)$ is any term between sets, the formula is defined as follows:

$$L\left(V_{t-k}V_{t-k+1}V_{t+k-1}|V_t\right) \tag{1}$$

Among them: L $(V_{t-k}V_{t-k+1}V_{t+k-1}|V_t)$ is the chance of $V(t)$ showing up by jumping combination. The n-dimensional vector expression of any word is output when the word vector trains on text:

$$V = \begin{bmatrix} V_{11} & V_{12} & V_{13} & \cdots & V_{1n} \\ V_{21} & V_{22} & V_{23} & \cdots & V_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{k1} & V_{m2} & V_{m3} & \cdots & V_{mn} \end{bmatrix} \tag{2}$$

LSI stands for the text set as an $m \times n$-dimensional matrix $|X|$, where m is the size of the dictionary, n is the number of texts, and the element (i, j) of the matrix is the rate of the ith word in the jth text. The matrix XX^T stands for the relationship of any two words. Singular-value decomposition of the matrix X:

$$X = U\Sigma V^T \tag{3}$$

U and V represent orthogonal matrices, and $\Sigma = \mathrm{diag}\,(\sigma_1, \sigma_2, \ldots, \sigma_l), XX^T$, and X^TX can be expressed as:

$$XX^T = \left(U\Sigma V^T\right)\left(U\Sigma V^T\right)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T \tag{4}$$

$$X^TX = \left(U\Sigma V^T\right)^T\left(U\Sigma V^T\right) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T \tag{5}$$

When the LSI model is completed, the output of the topic distribution matrix of any data in the dataset is as follows:

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & \cdots & E_{1n} \\ E_{21} & E_{22} & E_{23} & \cdots & E_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ E_{m1} & E_{m2} & E_{m3} & \cdots & E_{mn} \end{bmatrix} \tag{6}$$

$$V_{new} = V \oplus E \tag{7}$$

$\oplus$ is a combination of two matrices. The vector addition results in a fresh matrix V_{nev}. The vector superposition matrix is used as the starting point of the classifier model.

$$V_{new} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & \cdots & V_{1n} \\ V_{21} & V_{22} & V_{23} & \cdots & V_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ V_{k1} & V_{k2} & V_{k3} & \cdots & V_{kn} \\ V_{m1} & V_{m2} & V_{m3} & \cdots & V_{mn} \end{bmatrix} \tag{8}$$

3.2.2. Extraction Processing

Keywords are the representatives of text features. We used the LSI-Word2vec mapping method to extract features from the preprocessed files. The specific process is shown in Figure 3. The topic model is established according to LSI, computing TF-IDF values on the basis of word frequency and the properties of words, and the dimension is reduced by SVD, finally forming a vector model.

The candidate keyword set is extended according to the Word2vec model, and all words are mapped to a spatial vector of low latitudes according to the semantic relationship of the context.

The two matrices obtained by superposition training establish the semantic feature matrix, which can effectively improve the text expression method.
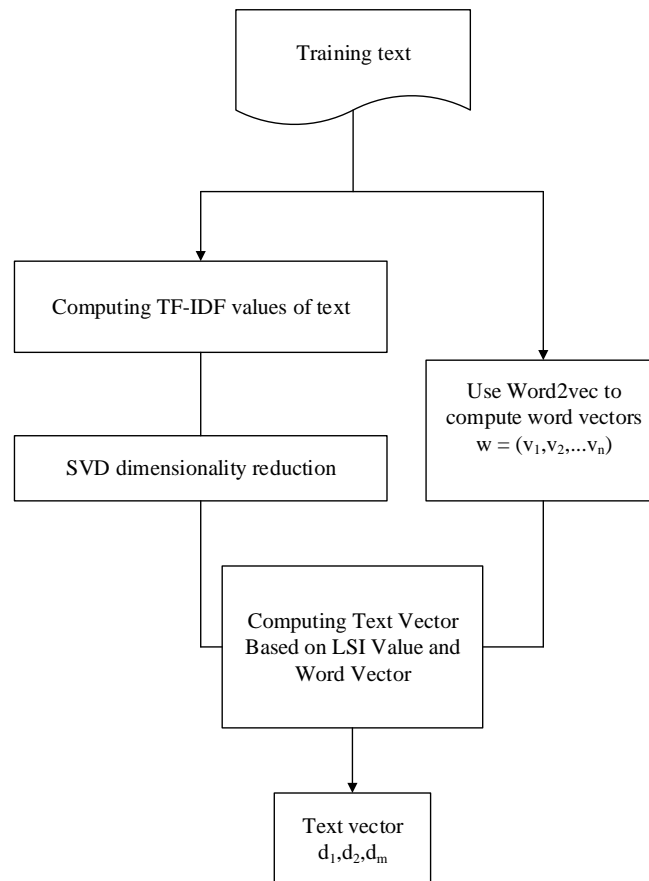
Figure 3. Feature extraction. TF-IDF, Term Frequency-Inverse Document Frequency.

3.3. Model Fusion

3.3.1. Pruning the Decision Tree

In order to simplify the stacking model, the decision tree pruning strategy was used. The main idea was to first prune a leaf node. The loss functions values of decision trees before and after pruning were compared. Then, the global optimal pruning scheme could be obtained through dynamic programming, as showed in Figure 4. When the expected error rate of a single leaf node was lower than the expected error rate of the corresponding subtree root node, the leaf node was used instead of the whole subtree.
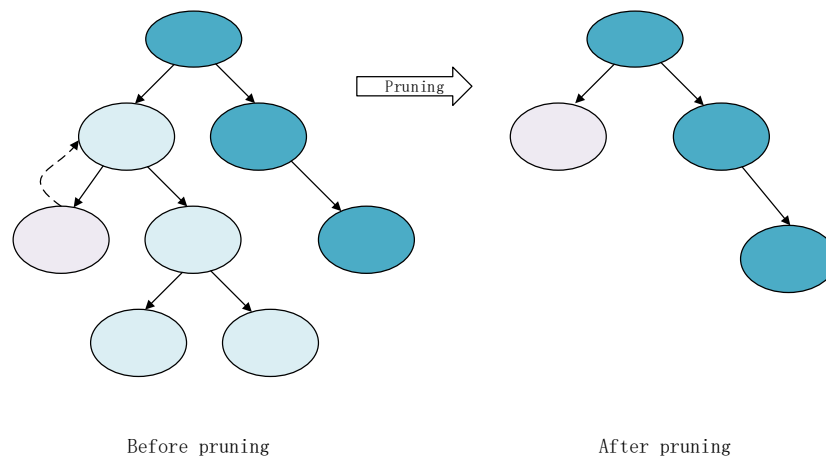

Figure 4. Decision tree pruning.

The Algorithm 1 is shown below. Let A_{mn} be a matrix of continuous attributes, with n instances and m attributes. Set the parent node and left and right nodes as t, t1, and t2 and gain as the loss function values.

Algorithm 1 The algorithm of decision tree pruning.

Require: A_{mn}: matrix of continuous attributes;
 1: **procedure**
 2: PruningTree(DT, δ);
 3: For each(t,t1,t2) $\in$ DT;
 4: **while** each(t,t1,t2) $\in$ DT **do**

 5: 　　Gain(t,t1,t2) $\in$ DT $< \delta$ then;
 6: 　　Trim t1,t2;
 7: 　　PruningTree(DT, δ);
 8: **end while**
 9: Return DT;
10: **endprocedure**

3.3.2. Stacking

On the premise of short computation time and high precision, the ensemble learning strategy was adopted to combine the classifiers. As is shown in Figure 5, in the first layer, we cross-verified the decision tree and other basic models to train multiple basic learners from multiple different feature sets. Model 1 was utilized to train the small training set and predict the folded small test set. The obtained results were input into the second layer as the training source of the second layer model. The second classifier combined the features of the best-performing classifiers to learn and fused them to obtain the final results.
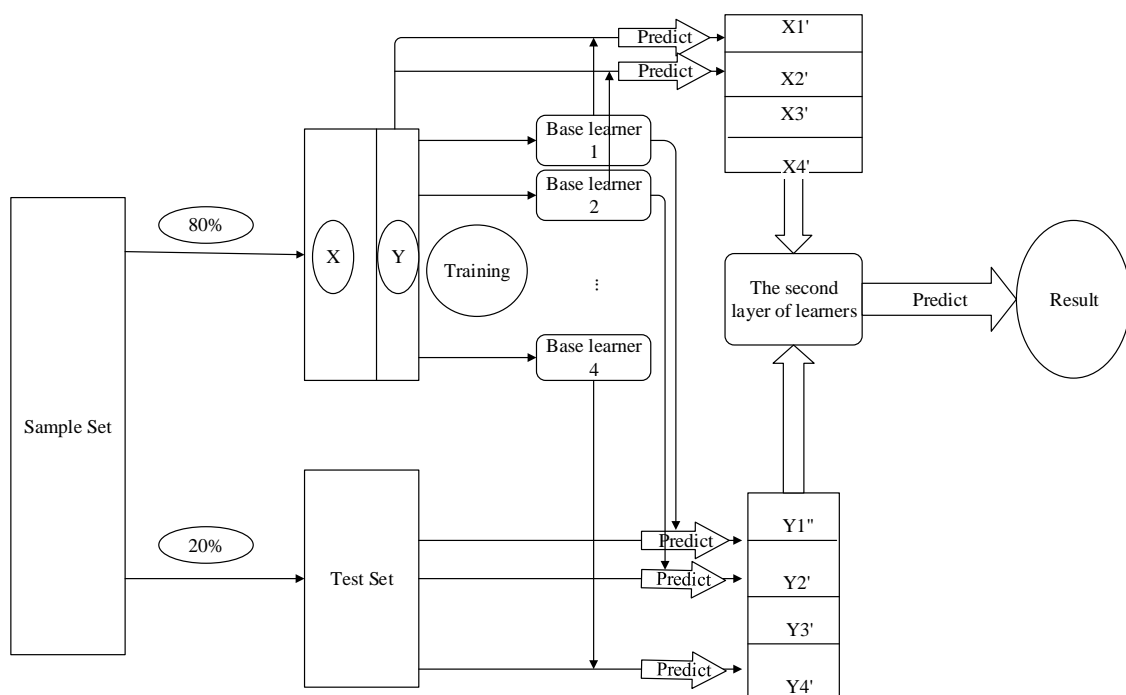

Figure 5. Model fusion.

4. Experiment Settings

4.1. Experimental Setup

The computer environment of the experiment was Windows 10 $\times$ 64 home edition. The CPU was Intel (TM) i5-8500 CPU @ 3.00 GHz. The capacity of memory was 8 GB. The environment of the

experiment was based on Anaconda Python 3.6, and the machine learning framework was sklearn 2.2.4. The dataset (native HTML format) we used came from SOHU's second content recognition algorithm competition [25]. In this competition, the dataset size was 50,000 pieces of tagged news records. We randomly used 40,000 pieces of the data for model training, 20,000 pieces of positive samples representing non-marketing news, and 20,000 pieces of negative samples representing marketing news. Eighty percent of the news had the function of training, and the rest had the function of testing to evaluate the effectiveness of the algorithm. The data file also contained properties as shown below.

4.2. Experimental Parameters

In this experiment, the parameters were optimized, and the most suitable parameters were selected. The parameters of the LSI-Word2vec used are shown in Table 1.

Table 1. The model parameters of LSI-Word2vec.

Corpus	Num_Topics	id2Word	Chunksize	Decay	Size	Sentence	min_Count
corpus_tfidf	2	dictionary	2000	1	50	corpus	5

The parameters of the stacking model used are shown in Table 2.

Table 2. The model parameters of Stacking.

Model	max_Depth	n_Estimators	Num_Leaves	cv	min_Samples
Gradient Boosting	13	100	-	-	-
Random Forest	-	32	-	-	2
XGBoost	3	-	-	-	-
LightGBM	-	2000	32	-	-
Stacking	-	-	-	5	-

4.3. Metrics

In the experiment, F1, Accuracy (Acc), and the Logarithmic Loss Function (Loss) were used as L = logarithmic Loss indicators. F1 is the weighted harmonic average of Precision (P) and Recall (R). F1 is an appropriate method to measure the reliability of the model. Accuracy represents the ratio of the number of classified correctly to the total number of the dataset. Loss is used to evaluate the inconsistency between the predicted result and the real result of the model. It is a function of non-negative real value. The smaller the Loss function is, the better the robustness of the model will be. The calculation formulas are shown as follows., where Y indicates the variable of the output variable, X the input variable, L the Loss function, N the sample size of the input, while y_i is the real category of input instance, and p_i the probability that the input instance belongs to Category 1.

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$P = \frac{TP}{TP + FP} \tag{10}$$

$$F1 = \frac{2PR}{P + R} \tag{11}$$

$$ACC = \frac{(TP + TN)}{P + N} \tag{12}$$

$$L(Y, P(Y|X)) = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log p_i + (1 - y_i) \log (1 - p_i)) \tag{13}$$

Table 3 shows the nature of three various feature extractions on account of the same dataset and the same classifier. By comparing the indicators of Acc and F1, it can be seen that LSI-Word2vec can fully extract semantic information to achieve a better classification effect.

Table 3. Comparison of the results of different feature extraction methods. LDA, Latent Dirichlet Allocation.

Model	Acc	F1
LSI + XGBoost	0.840	0.722
LDA + XGBoost	0.836	0.698
Word2Vec + XGBoost	0.832	0.694
LSI-Word2vec + XGBoost	0.848	0.739

In order to find the most suitable dimension of the method of feature extraction, the methods were compared according to different word vector dimensions under different feature extraction methods. The following is a comparison of the experimental results based on the stacking model in different dimensions, using different feature extraction methods. From Figure 6, it can be seen that the word vector dimension cannot be too low, and it is more appropriate to choose 100 or so.
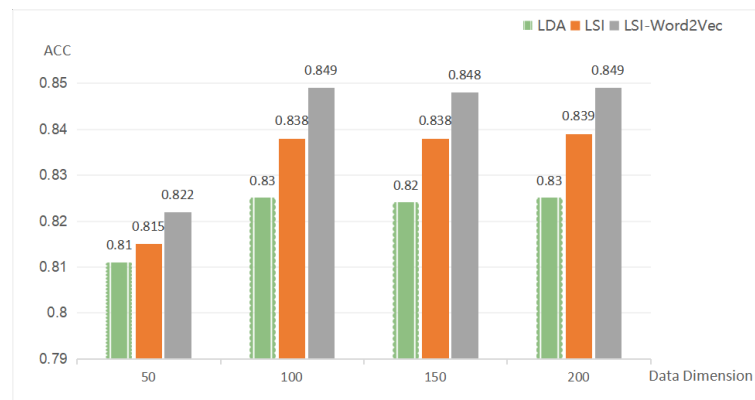

Figure 6. Comparison of the results from different dimensions.

The following is an experiment based on the feature extraction method of LSI-Word2Vec. The news recognition model included the results of the first layer of various decision trees of the stacking model and the results of the second layer integrating the models. The pruning-stacking model represents the result of post-pruning integration of decision trees.

The experimental results are as follows in Table 4. The stacking integration strategy can greatly enhance the performance of the classifier after multiple trainings. By comparing the running time of the unpruned decision tree model in the first layer of the stacking model to the running time of the decision tree after pruning, it can be seen that the computational time of the pruning strategy was greatly reduced. Compared with other classifier models, the model constructed in this paper can achieve better results. This model performed well in marketing intent recognition and can be executed more efficiently.

Table 4. Comparison of the classification results.

Model	F1	Acc	Loss	Time
Gradient Boosting	0.733	0.846	6.315	-
Random Forest	0.720	0.831	6.828	-
XGBoost	0.701	0.847	6.225	-
LightGBM	0.737	0.846	6.312	280
Stacking	0.739	0.848	6.222	220

5. Conclusions

The news text of the Internet is an important research object of natural language processing. Keywords extraction is a basic task of text analysis, which has important research significance for text categorization. This paper combined the matrix that fully embodies the relationship between words and the matrix that embodies the semantic relationship. The new matrix was formed by adding two vectors, which can reflect not only subject features, but also theme features. Based on the method combining LSI and Word2Vec, firstly, the characteristics of the LSI model can be utilized to obtain the relevance of word meaning and theme and optimize the method of keyword feature extraction. In addition, combined with the method of the Word2Vec model, words can be transformed through similarity to improve the overall keyword feature extraction effect.

The improved classifier model solved the problem of a long training time, using the decision tree pruning strategy to simplify the stacking model and shorten the computation time. In conclusion, the proposed method was very good at identifying marketing news.

References

1. Ma, K.; Yu, Z.; Ji, K.; Yang, B. Stream-based live public opinion monitoring approach with adaptive probabilistic topic model. *Soft Comput.* **2018**, *23*, 7451–7470. [CrossRef]
2. Zhang, C.; Wang, X.; Yu, S.; Wang, Y. Research on Keyword Extraction of Word2vec Model in Chinese Corpus. In Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 6–8 June 2018; pp. 339–343.
3. Kalra, S.; Li, L.; Tizhoosh, H.R. Automatic Classification of Pathology Reports using TF-IDF Features. *arXiv* **2019**, arXiv:1903.07406.
4. Zhu, Z.; Liang, J.; Li, D.; Yu, H.; Liu, G. Hot Topic Detection Based on a Refined TF-IDF Algorithm. *IEEE Access* **2019**, *7*, 26996–27007. [CrossRef]
5. Kim, D.; Seo, D.; Cho, S.; Kang, P. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Inf. Sci.* **2019**, *477*, 15–29. [CrossRef]
6. Altszyler, E.; Sigman, M.; Slezak, D.F. Corpus specificity in LSA and Word2vec: The role of out-of-domain documents. *arXiv* **2017**, arXiv:1712.10054.
7. Anandarajan, M.; Hill, C.; Nolan, T. Semantic Space Representation and Latent Semantic Analysis. In *Practical Text Analytics*; Springer: Cham, Switzerland, 2019; pp. 77–91.
8. Rajalakshmi, R.; Aravindan, C. A Naive Bayes approach for URL classification with supervised feature selection and rejection framework. *Comput. Intell.* **2018**, *34*, 363–396. [CrossRef]
9. Zhao, Z.; Wang, X.; Wang, T. A novel measurement data classification algorithm based on SVM for tracking closely spaced targets. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 1089–1100. [CrossRef]
10. Khaleel, M.I.; Hmeidi, I.I.; Najadat, H.M. An automatic text classification system based on genetic algorithm. In Proceedings of the the 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016, Union, NJ, USA, 15–17 August 2016; ACM: New York, NY, USA, 2016; p. 31.

11. Ding, X.; Shi, Q.; Cai, B.; Liu, T.; Zhao, Y.; Ye, Q. Learning Multi-Domain Adversarial Neural Networks for Text Classification. *IEEE Access* **2019**, *7*, 40323–40332. [CrossRef]

12. Narayanan, A.; Shi, E.; Rubinstein, B.I. Link prediction by de-anonymization: How we won the kaggle social network challenge. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 1825–1834.

13. Pavlyshenko, B. Using Stacking Approaches for Machine Learning Models. In Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 21–25 August 2018; pp. 255–258.

14. Zou, H.; Xu, K.; Li, J.; Zhu, J. The Youtube-8M kaggle competition: Challenges and methods. *arXiv* **2017**, arXiv:1706.09274.

15. Liu, J.; Shang, W.; Lin, W. Improved Stacking Model Fusion Based on Weak Classifier and Word2vec. In Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 6–8 June 2018; pp. 820–824.

16. Kim, K.H.; Park, Y.M.; Ryu, K.R. Deriving decision rules to locate export containers in container yards. *Eur. J. Oper. Res.* **2000**, *124*, 89–101. [CrossRef]

17. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.

18. Gao, X.; Luo, H.; Wang, Q.; Zhao, F.; Ye, L.; Zhang, Y. A Human Activity Recognition Algorithm Based on Stacking Denoising Autoencoder and LightGBM. *Sensors* **2019**, *19*, 947. [CrossRef] [PubMed]

19. Wang, J.; Lou, C.; Yu, R.; Gao, J.; Xu, T.; Yu, M.; Di, H. Research on Hot Micro-blog Forecast Based on XGBOOST and Random Forest. In *International Conference on Knowledge Science, Engineering and Management*; Springer: Cham, Switzerland, 2018; pp. 350–360.

20. Xi, Y.; Zhuang, X.; Wang, X.; Nie, R.; Zhao, G. A Research and Application Based on Gradient Boosting Decision Tree. In *International Conference on Web Information Systems and Applications*; Springer: Cham, Switzerland, 2018; pp. 15–26.

21. Li, Y.; Yan, C.; Liu, W.; Li, M. A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification. *Appl. Soft Comput.* **2018**, *70*, 1000–1009. [CrossRef]

22. Sun, J. *'Jieba' Chinese Word Segmentation Tool*; Gitlab: San Francisco, CA, USA, 2012.

23. Xu, Y.; Wang, J. The Adaptive Spelling Error Checking Algorithm based on Trie Tree. In Proceedings of the 2016 2nd International Conference on Advances in Energy, Environment and Chemical Engineering (AEECE 2016), Singapore, 29–31 July 2016; Atlantis Press: Paris, France, 2016. [CrossRef]

24. Liu, G.; Kang, Y.; Men, H. CHAR-HMM: An Improved Continuous Human Activity Recognition Algorithm Based on Hidden Markov Model. In *Mobile Ad-hoc and Sensor Networks: 13th International Conference, MSN 2017, Beijing, China, 17–20 December 2017*; Revised Selected Papers; Springer: Berlin, Germany, 2018; Volume 747, p. 271.

25. Zecheng Zhan SOHU's Second Content Recognition Algorithm Competition. Available online: https://github.com/zhanzecheng/SOHU_competition (accessed on 29 May 2019).