An Improved Class-center Method for Text Classification Using Dependencies and WordNet

Xinhua Zhu[1#], Qingting Xu[1#], Yishan Chen[1, 2,*], Tianjun Wu[1]
1. Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, China
2. International Business School, Guilin Tourism University, Guilin, China
zxh429@263.net, xat341126@163.com, GLCYS@163.com[*], 1035626809@qq.com

Abstract. Automatic text classification is a research focus and core technology in natural language processing and information retrieval. The class-center vector method is an important text classification method, which has the advantages of less calculation and high efficiency. However, the traditional class-center vector method for text classification has the disadvantages that the class vector is large and sparse; its classification accuracy is not high and it lacks semantic information. To overcome these problems, this paper proposes an improved class-center method for text classification using dependencies and the WordNet dictionary. Experiments show that, compared with traditional text classification algorithms, the improved class-center vector method has lower time complexity and higher accuracy on a large corpus.

Keywords: Text Classification, Dependency, Weight Calculation, WordNet, Class-center Vector.

1 Introduction

With the rapid development of Internet technology, network information has exploded in an exponential manner. How to effectively organize and manage this text information becomes an urgent problem to be solved [1]. Text classification is one of the important research directions [2].

Common text classification algorithms include the Bayesian classification [3], *K*-nearest neighbor (KNN) [4], support vector machine (SVM) [5], and class-center vector algorithms [6]. Although the Bayesian algorithm is simple in principle and easy to implement, it is based on the hypothesis that the classification accuracy will be high only if the text dataset is independent of each other [7]. The classification accuracy of KNN is very high, but the classification efficiency is very low. SVM is widely used in small corpora because of its strong generalization ability, but it is not very effective in large corpora [8]. The main advantage of the class-center vector method is that the corpus is greatly reduced before its classification process [9].

# Joint first author Xinhua Zhu and Qingting Xu, email addresses: zxh429@263.net, xat341126@163.com

* Corresponding author Yishan Chen , email addresses: GLCYS@163.com

Therefore, its classification process has a less calculation and high classification efficiency. However, the traditional class-center vector algorithms for text classification have the disadvantages that the class vector is large and sparse; classification accuracy is not high and lacks semantic information.

In terms of weight calculations for text vectors, in 1973, Salton et al. [10] combined the idea of Jones [11] to present a TFIDF (Frequency & Inverse Documentation Frequency Term) algorithm. The TFIDF algorithm has been highly favored by the relevant researchers [12-14] and many application fields, because of its easy understanding, simple operation, low time complexity, high accuracy and high recall rate. To further improve its performance, scholars have made continuous efforts. For example, How and Narayanan [12] put forward the Category Term Descriptor (CTD) to improve TFIDF in 2004. It solved the adverse effect of the number of documents in different categories on the TFIDF algorithm. Qu et al. [13] proposed a new approach for calculating text vector weights, which combined simple distance vector to traditional TFIDF algorithms and obtained the very good classification effect. In 2012, Wang et al. [14] proposed a new TFIDF algorithm based on information gain and information entropy. This method only considers the feature words with high information gain. The above methods have made some progress in improving the performance of TFIDF algorithm, but they all lack the combination of text semantics to understand the text content.

Principal Component Analysis (PCA) [15] and Non-negative Matrix Factorization (NMF) [16] are traditional techniques for dimensionality reduction. However, the PCA contains both positive and negative values in the decomposed matrices, the cost of PCA computation will be prohibitive when matrices become large. The NMF is distinguished from the PCA method by its non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. Besides, PCA and NMF are only suitable for vectors have the same order of magnitude, and both require dimensionality reduction. In this paper, the dimension of the class-center vector is much bigger than the classified text vector, and the order of magnitude is not equivalent. Therefore, neither PCA nor NMF is suitable for the dimensionality reduction of this paper.

To overcome the above problems, this paper proposes an improved class-center vector method for text classification based on dependencies and the semantic dictionary WordNet. The main contributions of this paper can be summarized as follows:

(1) Aiming at the semantic defects of the statistics-based TFIDF algorithm, we introduce dependencies and the synonyms in the WordNet dictionary to understand and optimize the text feature, and put forward an improved weight calculation algorithm based on TFIDF.

(2) We use the category nodes located in the 6-9 layers of WordNet to cluster feature words in the class-center vector and to significantly reduce the dimension of class-center vector, thereby realizing a new class-center vector for text classification using dependencies and the WordNet dictionary.

(3) Since the dimension of our clustered class-center vector is very different from that of the classified text vector, the similarity between them is not suitable to directly use the traditional cosine similarity method. This paper proposes a new

vector similarity method for our clustered class-center vector, in which the similarity between the class-center vector and the classified text vector is expressed as the ratio of the sum of the classified text feature weights matching with the class center vector and the sum of all the weights of the class center vectors. It can improve the accuracy of our class-center vector text classification.

2 Class-center Vector Method

The basic idea of the class-center vector method [6] is to use the arithmetic average method to determine the class-center vector of each class, calculate the similarities between the classified text vector and each class-center vector according to the cosine similarity formula, and assign the classified text into the category with the highest similarity value. The detailed calculation steps are as follows:

(1) The arithmetic average formula is used to determine the class-center vector. The formula is as follows:

$$\mathbf{v}_{C_k} = \left\{ \left(t_{k,j}, w_{k,j}\right) \middle| \, j \in \{1,2,\ \cdots,\ m\} \text{ and } w_{k,j} = \frac{1}{S_k}\sum_{i=1}^{S_k} w_{k_{i,j}} \right\} \tag{1}$$

where, m is the feature dimension of class-center vector; $t_{k,j}$ represents the jth feature of the class-center vector of the kth class; $w_{k,j}$ is the weight value of the jth feature of the class-center vector of the kth class; $S_k$is the total number of text in the kth category in the training set; $w_{k_{i,j}}$ represents the weight value of the jth feature of the ith text in the kth category, and can be calculated by a feature weight algorithm (such as the TFIDF algorithm).

(2) The xth classified text is represented as a text feature vector $\mathbf{v}_{d_x}$:

$$\mathbf{v}_{d_x} = \left\{ \left(t_{x,j}, w_{x,j}\right) | \, j \in \{1,2,\ \cdots,\ l\} \right\} \tag{2}$$

where, l is the dimension of the text feature vector; $t_{x,j}$ denotes the jth feature of the xth classified text; $w_{x,j}$ is the weight value of the jth feature in the xth classified text, and can be calculated by the feature weight algorithm.

(3) Cosine similarity is generally used to calculate the similarity between the class-center vector and the classified text vector, and the formula is as follows:

$$Sim(\mathbf{v}_{C_k}, \mathbf{v}_{d_x}) = \frac{\left\langle \mathbf{v}_{C_k}, \mathbf{v}_{d_x} \right\rangle}{\left\| \mathbf{v}_{C_k} \right\| \times \left\| \mathbf{v}_{d_x} \right\|} \tag{3}$$

(4) All the calculated similarity values are sorted by their values, and the classified text is classified into the category with the largest similarity value.

3 Proposed Method

3.1 Preprocessing

To perform a text classification experiment, we first need to convert the text in the corpus into a form of data that the computer can directly process, and the preprocessing is the first step to complete the transformation. The preprocessing in this paper includes stemming and stop words deletion.

3.2 TFIDF Weight Improvement Based on Dependencies and WordNet

Syntactic analysis based on dependencies can reflect the semantic relationship between the components in a sentence, and is not affected by the physical location of the component [17]. Now it is widely used in the analysis of sentence structure. Firstly, according to the different dependencies between the word and the predicate in sentences, we determine the importance of the word to the sentence, the text and even the category, that is, determines the importance of the word to the text according to the sentence component represented by the word. Then, according to the importance of different components to the sentence, we divide the sentence components into eight levels (see Table 1), and propose an improved TFIDF method for text classification according to Table 1.

Table 1. Dependency levels

Sentence components	dependencies	level
subject	Subj (subject)	1
	Nsubj (noun subject)	
	Npsubj (passive subject)	
	Nsubjpass (passive noun subject)	
object	Obj (object)	2
	Dobj (direct object)	
	Iobj (indirect object)	
nominal modifier	Nmod (compound noun modification)	3
	Npadvmod (noun as adverbial)	
predicate	Root (central word)	4
attribute	Assmod (correlation modification)	5
	Numod (quantitative modification)	
complement	Comp (complement)	6
	Acomp (adjective complement)	
	Tcomp (time complement)	
	Lccomp (location complement)	
adverbial	Advmod (adverbial)	7
other	Other dependencies	8

In a sentence, the subject, as the agent of the predicate, is the most important component, so this paper classifies the characteristics of all the subject components as the first level feature. As the object of the predicate, the object is the sub-important component, and the characteristics of all the object components are classified as the second level feature. Nominal modifiers are classified as the third level feature. Predicate is the core of a sentence, but it is generally a verb and it is a central word in the dependencies syntax. Verbs have the universal applicability, so they are not as important to text classification as nouns. Therefore, all the predicate component words are classified as the fourth level characteristic. The definite-middle relationship and adverbial-middle relationship are generally produced by adjectives and adverbs. As a sentence component, they may be the three major categories of attributive, complement, and adverbial, which are classified into the fifth, sixth and seventh levels. In addition, words such as Mod (modifier), Pass (passive modification), Tmod (time modification), Amod (adjective modification), and Advmod (adverb modification) are all classified as the eighth level feature.

After classifying the text features in the dataset according to dependencies, this paper proposes the following TFIDF weight calculation method based on dependencies and the synonyms in the WordNet dictionary. The specific steps are as follows:

(1) The synonyms in the text are merged according to the WordNet dictionary, in which the first word of the synonym group in the WordNet dictionary is used as a feature representation for all synonyms.

(2) We calculate the number of times that the feature word t_i appears in the text, which is set to m. Then, according to the result of dependency syntactic analysis implemented by Stanford Parser[2], we get the sentence component to which the feature word t_i belongs to its jth ($1 \leq j \leq m$) occurrence in the text, and classify the jth occurrence of the feature t_i in the text as the $k_{i,j}$ level according to Table 1 and assigns it a weight $w_{i,j}$, which is calculated as follows:

$$w_{i,j} = 2\cos\left[\left(\frac{k_{i,j}}{8}\right)^{\lambda} \times \frac{\pi}{2}\right] \tag{4}$$

where λ is a parameter, which is used to adjust the weight gap between feature grades, and its range is [0, 1];

(3) The improved frequency TF_i with weights for the feature word t_i in the text is calculated as follows:

$$TF_i = \sum_{j=1}^{m} w_{i,j} = \sum_{j=1}^{m} 2\cos\left[\left(\frac{k_{i,j}}{8}\right)^{\lambda} \times \frac{\pi}{2}\right] \tag{5}$$

(4) Finally, we propose the following improved TFIDF weight formula based on dependency and WordNet for feature word t_i:

$$TF\_IDF_i = \frac{\sum_{j=1}^{m} 2\cos\left[\left(\frac{k_{i,j}}{8}\right)^{\lambda} \times \frac{\pi}{2}\right]}{s} \times \log\left(\frac{D}{p_i} + 0.01\right) \tag{6}$$

where s denotes the total number of words in the text where feature t_i is located and D denotes the total number of texts in the dataset, p_i denotes the number of

[2] https://nlp.stanford.edu/software/lex-parser.html

the texts containing the feature t_i.

3.3 Class-center Vector Clustering Approach Based on WordNet

In the traditional class-centric method, the dimension of a class vector is the union of all the text vectors of the class in the training set, which is very large and sparse. Therefore, the classification accuracy of traditional class-centric methods is not very high. Although, WordNet-based synonym merging can reduce the dimension of the class-center vector to some extent, this is far from enough. To effectively reduce the dimension of the class-center vector, we use the taxonomic hierarchy in WordNet to cluster the feature words of the class-center vector.

WordNet [18] is a large semantic dictionary based on cognitive linguistics and is designed and realized by psychologists, linguisticians and computer engineers in Princeton University. Considering that the average depth of the WordNet taxonomy reaches 10 layers, we use the category nodes in the first to ninth layers of the WordNet taxonomy to perform clustering effect test on the 20Newsgroups corpus, and the experimental results are shown in Figure 1 below.

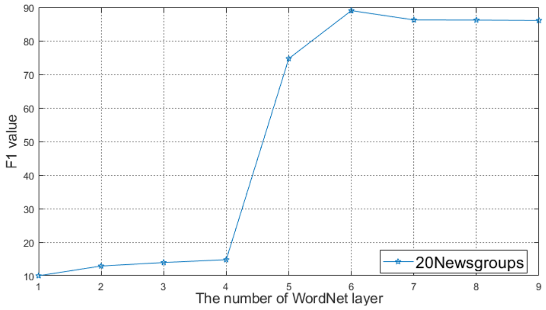

Fig. 1. Influence of the number of WordNet layers on F_1 value

In Figure 1, when features are clustered to the first to fourth layers in WordNet, because the category nodes in the first to fourth layers of the WordNet are too abstract, all the features are grouped to the top and the abstract hypernym features in WordNet, so the classification effect is very poor. When the feature is clustered to the sixth layer of WordNet, the classification effect achieves the best, in which the F_1 value reaches 88.97%.when features are clustered to the seventh to ninth layers , the classification effect is still good although it has decreased. Therefore, the coding of the synonym sets of largest common subsumes located in the 6-9 layers of WordNet is used as the clustering feature. The specific clustering process is as follows:

Firstly, the initial value of the class-center vector is determined by the arithmetic average of the weight of the feature in all documents of the class. The formula is as follows:

$$V_{C_k^0} = \{(t^0_{k,j}, w^0_{k,j}) \mid j \in [1, L] \text{and}\, w^0_{k,j} = \frac{1}{S_k}\sum_{i=1}^{s_k} w_{k_{i,j}}\} \tag{7}$$

where $V_{C_k^0}$ represents the initial class-center vector of the *k*th category; *L* is the dimension of the initial class-center vector; $t^0_{k,j}$ represents the *j*th feature in the initial class-center vector of the *k*th category; $w^0_{k,j}$ is the initial weight value of the *j*th feature in the initial class-center vector of the *k*th category; S_k represents the total number of the texts of the category *k* in the training set, $w_{k_{i,j}}$ represents the weight value of the *j*th feature in the *i*th text of category *k*.

Then, the dataset is clustered through the WordNet dictionary. If the level of the arbitrary initial features $t^0_{k,j}$ in the WordNet is less than or equal to 6, the coding of its synonym group in WordNet is used as its clustering feature. Otherwise, we use the coding of the synonym set of its largest common subsume located in the 6-9 layers of WordNet as its clustering feature. The largest common subsume is the least common subsume that is located in the 6th to 9th layer of WordNet and contains the most characteristic words in the given initial vectors, such as the *b* node in Fig. 2.

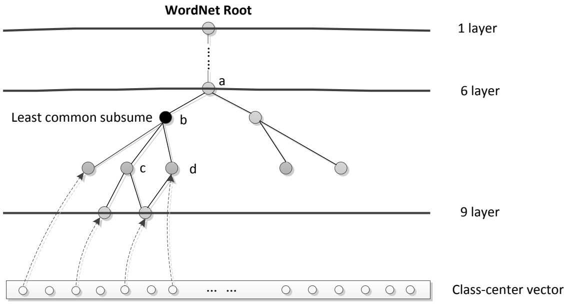

Fig. 2. An instance of the least common subsume in the clustering process

Finally, all the features of the initial class center vector of the *k*th category are clustered according to the above steps, and then, according to the following formula, the clustered center vector of the *k*th category is obtained.

$$V_{C_k} = \{(T_{k,j}, W_{k,j}) \mid j \in [1, n] \text{ and } W_{k,j} = \sum_{t^0_{k,i} \to T_{k,j}} w^0_{k,i}\} \tag{8}$$

where V_{C_k} represents the clustered center vector of the *k*th class, *n* is the dimension of the clustered center vector and *n* is less than or equal to the initial dimension *L* of the class-center vector. $T_{k,j}$ denotes the *j*th feature of the *k*th class after clustering, $W_{k,j}$ is the weight of $T_{k,j}$, $\sum_{t^0_{k,i} \to T_{k,j}} w^0_{k,i}$ represents the sum of weights for all the initial features that participate in the $T_{k,j}$feature clustering.

3.4 A New Vector Similarity Method for Clustered Class-center Vectors

Since the dimension of our clustered class-center vector is very different from that of the classified text vector, the similarity between them is not suitable to directly use the traditional cosine similarity method. This paper proposes a new vector similarity method for our clustered class-center vector, in which the similarity between the class-center vector and the classified text vector is expressed as the ratio of the sum of the classified text feature weights that is matched with the class center vector and the sum of all the weights of the class center vectors. The specific calculation processes are as follows:

(1) According to the dependency-based feature selection method and the improved TFIDF calculation method for the feature weight, the clustered class-center vector V_{C_k} for the category C_k and the feature vector V_{d_x} for the classified text $d_x$are determined;

(2) The V_{C_k}, $V_{d_x}$are inversely sorted by weights, and the first θ weights are taken. The calculation formula is as follows:

$$V_{C_k}^{\theta} = \{(T_{k,j}, W_{k,j}) \mid j \in [1, \min(\theta, n)] \text{ and } W_{k,j} \geq W_{k,j+1}\} \tag{9}$$

$$V_{d_x}^{\theta} = \{(t_{x,j}, w_{x,j}) \mid j \in [1, \min(\theta, q)] \text{ and } w_{x,j} \geq w_{x,j+1}\} \tag{10}$$

where θ represents a range of values from 0 to 3000, that is, selecting the most suitable dimension for the vectors V_{C_k}, V_{d_x} can make the classification effect the best, $V_{C_k}^{\theta}$represents the class-center vector of the kth class with the θ dimension, $V_{d_x}^{\theta}$ denotes a feature vector of the classified text d_x with the θ dimension, n and q represent the initial dimensions of the vectors V_{C_k}, V_{d_x}, respectively

(3) We propose a new formula to calculate the similarity between the feature vector $V_{d_x}^{\theta}$of the classified text d_x and the clustered class-center vector $V_{C_k}^{\theta}$ of the kth class as follows:

$$Sim(V_{C_K}^{\theta}, V_{d_x}^{\theta}) = \frac{\sum_{t_i \in Stem(d_x \to C_K)} W_{C_k}(t_i)}{\sum_{t_i \in Stem(C_K)} W_{C_k}(t_i)} \tag{11}$$

where Stem(C_k) denotes the feature set in vector $V_{C_k}^{\theta}$, Stem($d_x \to C_k$) represents a feature set in the class-center vector $V_{C_k}^{\theta}$ that can be successfully matched by the features in the classified text d_x. For any feature $t_{x,i}$ in the classified text d_x, the match rule between it and any $T_{k,j} \in$ Stem(C_k) is as follows: if $t_{x,i}$ and $T_{k,j}$ have the same encoding in WordNet or $t_{x,i}$ belongs to the hyponym of $T_{k,j}$ in the WordNet taxonomy, then $t_{x,i}$ successfully matches with $T_{k,j}$; otherwise, they are mismatch.

4 Experiments and Analysis

In this paper, we used a popular 20Newsgroups[3] dataset as experimental corpus. 20Newsgroups is composed of 20 categories with a total of 19997 texts, in which each text is an article about a certain category. Because the articles in the corpus are moderate in length and grammatical, these articles are very suitable for dependency analysis. In our experiments, 20Newsgroups is randomly separated into a training set

[3] http://qwone.com/~jason/20Newsgroups

and a test set according to the ratio of 9:1. After comparing the optimized experiments on 20Newsgroups, we discovered that the best value of θ in Eqs. (9) and (10) is 3000. The computer configuration used in the experiment is: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz Memory 8G.

4.1 Comparison of Improved TFIDF Weight on Different Classification Methods

To verify the universality of our proposed TFIDF weight improvement approach based on dependencies and WordNet synonyms, we combined the improved TFIDF weight with the Bayesian, KNN and class-center classification methods on the 20Newsgroups dataset to evaluate its superiority. The improvements of F_1 values on different classification methods are shown in Table 2:

Table 2. Improvement of F_1 value using improved weight method

Weight approach	Classification method	F_1 improvement (rate %)
Dependencies	Bayesian	83.21 to 86.02 (3.38)
	KNN	78.28 to 84.15 (7.50)
	Class-center	78.26 to 82.72 (5.70)
Dependencies + WordNet synonyms	Bayesian	83.21 to 86.88 (4.41)
	KNN	78.28 to 85.83 (9.64)
	Class-center	78.26 to 83.74 (7.00)

It can be discovered from Table 2 that dependencies contributes the most to the improvement of the TFIDF weight, in which the dependency-based TFIDF weight improves the F_1 value of Bayesian classification from 83.21% to 86.02% (improvement rate=3.38%), the F_1 value of KNN classification from 78.28% to 84.15% (improvement rate=7.50%) and the F_1 value of class-center classification from 78.26% to 82.72% (improvement rate=5.70%). After the introduction of WordNet synonyms, our proposed TFIDF weight approach can further improve the effects of various classification methods, which shows that the introduction of WordNet synonyms in the TFIDF weight calculation has a certain degree of contribution to classification accuracy. Overall, our TFIDF weight approach can improve the Bayesian classification by 4.41%, the KNN classification by 9.64% and the class-center classification by 7%, which shows that our TFIDF weight approach is effective for various classification methods.

4.2 Comparison of Three Innovation Points on the Class-center Method

In this paper, we propose three innovation points: a TFIDF weight improvement approach, a class-center vector clustering approach and a new vector similarity algorithm. To better reveal the role these innovations play in the proposed classification method on the 20Newsgroups dataset, we overlay each innovation point one by one to the original class-centric classification method. The experimental results are shown in Table 3.

Table 3. Comparison of improved class-center method and the original methods

KNN method		Original class-center method		Class-center+ improved weight		Improved weight + clustering approach		Improved weight+ clustering + new similarity	
time	F_1	time	F_1	time	F_1	time	F_1	time	F_1
1h55min	78.26%	20s	78.26%	20s	83.74%	18s	86.01%	15s	88.97%

Table 3 shows that our improved class-center method significantly improves the F_1 value of the original class-center classification from 78.26% to 88.97% (improvement rate=13.68%), in which the proposed TFIDF weight approach improves the F_1 value of the original class-center classification from 78.26% to 83.74% (improvement rate=7%), the proposed class-center vector clustering approach further improves the F_1 value of the class-center classification from 83.74% to 86.01% (improvement rate=2.9 %) and the proposed class-center vector clustering approach further improves the F_1 value of the class-center classification from 86.01% to 88.97% (improvement rate=3.78%). Moreover, our improved class-center method significantly reduces the classification time of the KNN method from 1 hour 55 minutes to 15 seconds.

4.3 Comparison of Our Improved Method with Various Classification Methods

To verify the superiority of our improved class-center method in terms of performance, we compared our improved class-center method with various classification methods on the 20Newsgroups dataset, including with the KNN, SVM, Bayesian, 2RM (A method of two-level representation model based on syntactic information and semantic information) and original class-center classification methods. The experimental results are shown in Table 4.

Table 4. Comparison of different classification methods on 20Newsgroups

Classification method	F_1 value (%)	Evaluation in
KNN-based method [19]	78.28	[19]
SVM-based method [19]	84.85	[19]
Bayesian-based method [20]	83.21	[20]
2RM method [21]	83.25	[21]
Original class-center method [22]	78.26	[22]
Our class-center method	88.97	This work

Table 4 shows that our improved class-center method is superior to the current popular classification methods such as KNN, SVM, Bayesian and 2RM in classification accuracy, especially to significantly improve the classification effect of the KNN and class-center vector methods, which benefits from the following three aspects: (1) The dependency-based feature level makes the TFIDF weight calculation more reasonable; (2) The feature word clustering based on WordNet effectively reduces the high dimension and sparsity of the class center vector; and (3) The vector similarity algorithm effectively solves the dimensional inconsistency between the class center vector and the classified text vector.

5 Conclusions

This study reveals: (1) semantic techniques such as dependency level and synonym combination can effectively improve the calculation of text weights based on statistics, and have better performance in various classification methods on the article corpus; (2) WordNet can play an important role in the clustering of text vectors; (3) targeted similarity algorithm can significantly improve the similarity between text vectors with inconsistent dimensions.

In the next step, we will apply our proposed method in this paper and the Chinese semantic dictionary HowNet[4] to Chinese text classifications, thereby further improving the efficiency and accuracy of Chinese text classifications.

Acknowledgements

This work has been supported by the Natural Science Foundation of Guangxi of China under the contract number 2018GXNSFAA138087, the National Natural Science Foundation of China under the contract numbers 61462010 and 61363036, and Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing.

References

1. SS Li, X R, CQ Zong, CR Huang.: A framework of feature selection methods for text categorization. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 2,692-700 (2009).
2. XL Deng, YQ Li, J Weng, JL Zhang.: Feature selection for text classification: A review. Multimedia Tools & Applications. 78(3), 3793-3816 (2018).
3. R Abraham, J B Simha, S S Iyengar.: Medical datamining with a new algorithm for feature selection and Naive Bayesian classifier. International Conference on Information Technology. 44-49, (2007).

[4] http://www.keenage.com/html/e_index.html

4. H Yigit.: A weighting approach for KNN classifier. International Conference on Electronics, Computer and Computation. 8, 228-131 (2014).
5. JL Awange, B Paláncz, RH Lewis, L Völgyesi.: Support Vector Machines (SVM). Tékhne, Revista de EST udos Politécnicos. (2018).
6. WW Cohen.: Context-sensitive learning methods for text categorization. Conference on Research & Development in Information Retrieval. 307-315(1996).
7. JN Chen, HK Huang, SF Tian, YL Qu.: Feature selection for text classification with Naive Bayes. Expert Systems with Applications. 36(3), 5432-5435(2009).
8. https://blog.csdn.net/amds123/article/details/53696027,last accessed 2019/5/17.
9. G mao.: Research and implementation of text Classification Model based on Class Center Vector. Dalian University of Technology. Dalian,(2010)
10. G Salton, CT Yu.: On the construction of effective vocabularies for information retrieval. Proceedings of the 1973 Meeting on Programming Languages and Information Retrieval. 48-60 (1973).
11. KS Jones.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. 28(1), 11-21(1972).
12. BC How, K Narayanan.: An empirical study of feature selection for text categorization based on term weightage. Web Intelligence, WI 2004.IEEE/WIC/ACM International Conference on. 599-602 (2004).
13. SN Qu, SJ Wang, Y Zou.: Improvement of text feature selection method based on TFIDF. IEEE Computer Society. (2008).
14. DX Wang, XY Gao, Andreae P.: Automatic keyword extraction from single sentence natural language queries. PRICAI 2013. 637-648 (2012).
15. H Abdi, L J Williams.: Principal component analysis. Wiley Interdisciplinary Reviews Computational Statistics. 2(4), 433-459(2010).
16. S Tsuge, M Shishibori, S Kuroiwa, et al.: Dimensionality reduction using non-negative matrix factorization for information retrieval. 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236). 2,960-965 (2001).
17. L.Tesiniere.: Elements de Syntaxe Structurale . Libairie C, Klincksieck. (1959).
18. X Zhu,Y Yang, Y Huang, Q Guo, B Zhang.: Measuring similarity and relatedness using multiple semantic relations in WordNet. Knowledge and Information Systems. First Online: 01 August 2019, https://doi.org/10.1007/s10115-019-01387-6 (2019).
19. GZ Feng, ST Li, TL Sun, BZ Zhang.: A probabilistic model derived term weighting scheme for text classification. Pattern Recognition Letters. 110 (1), 23-29(2018).
20. Y Liu, RC Huang.: Research on Optimization of maximum discriminant feature selection algorithm in text Classification. Journal of Sichuan University (Natural Science Edition). 56(1), 65-70(2019).
21. J Yun, L Jing, J Yu, et al.: A multi-layer text classification framework based on two-level representation model. Expert Systems with Applications. 39(2), 2035-2046(2012).
22. SJ Cao.: Fuzzy Support Vector Machine of Dismissing Margin Based on the Method of Class-center. Computer Engineering and Applications. 42(22), 146-149(2006).