Research on Knowledge Classification Based on KNN and Naive Bayesian Algorithms

Wu Chenyang1,a  Wang Xiaohua2,b  Wang Jinbo3,c  Song Jiawei4,d

1University of Chinese Academy of Sciences Beijing, China
Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences Beijing, China
2Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences Beijing, China
3Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences Beijing, China
4University of Chinese Academy of Sciences Beijing, China
Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences Beijing, China

a781711426@qq.com  bwangxiaohua@csu.ac.cn  cwangjinbo@csu.ac.cn  dsongjiawei17@csu.ac.cn

ABSTRACT: With the development of the Internet, the speed of knowledge update is getting faster and faster, and the Internet is full of various knowledge data. For knowledge management systems, the knowledge classification function is very important. The main task of this paper is to learn the mapping relationship between text content and text category according to the given text content and text category, so as to get a classification model. Finally, use this classification model to process the text of the unknown classification, and judge or predict the category of the unknown text.

1. BACKGROUND AND SIGNIFICANCE
With the vigorous development of the Internet, profound changes have taken place in the production and dissemination of content. Every day a great deal of knowledge would be created, which leads to the phenomenon called information overload in recent years. How to extract valuable information from mass knowledge efficiently and accurately is a problem worth to be studied. Text data, as a very important bearing form of knowledge data, widely exists in the Internet world. Therefore, natural language processing technology plays an important role in the Internet information age, and has been paid much attention in academia and industry. Text categorization is one of the research directions. It is widely used in information retrieval, information filtering, digital library, news classification and other fields. Text categorization refers to the use of machine learning or in-depth learning algorithm to train a large number of text data to obtain a text categorization model, and then use this model to predict the text category which unknown categories of documents belong to. Text categorization is considered to be the key technology to solve the problem of information clutter. It can effectively support the organization and management of knowledge data, facilitate users to accurately locate the information resources they need, and is of great significance for the efficient use and management of knowledge.
text categorization technology is applied to knowledge management system to provide text categorization services in knowledge management system[1].

2. RELEVANT RESEARCH

2.1 Research on Text Classification
Text categorization has a long history of research. Since the 1960s, text categorization mainly uses manual formulation of some criteria to classify texts. This method requires a lot of time and manpower, and in order to construct appropriate rules, having a certain understanding of the relevant fields of knowledge is necessary. In the 1990s, with the development of the Internet, the explosion of online E-TEXT resources and the prosperity of machine learning research ignited the enthusiasm of researchers. Text categorization has evolved into automatic categorization of unclassified new texts by establishing classification models or rules on training sets. A large number of experimental results show that its accuracy can match the manual classification of experts. In recent years, researchers have done a lot of research to improve the accuracy of classification.

Research on text categorization abroad is early. In 1971, Rocchio developed a method to construct a linear classifier based on class weight vectors, and the construction of vectors is the feedback information in the process of user retrieval. In 1979, some concepts proposed by Keith van Rijsbergen in information retrieval research were introduced into text classification, such as Vector Space Model, evaluation measures such as Recall Rate and Accuracy Rate. The probability model proposed by him also was adopted by most researchers in the future. In 1992, Lewis elaborated the detailed methods used in the implementation of a text categorization system, and experimented on the Reuters 22713 data set created by him. It gave people a comprehensive understanding of the field of text categorization and provided a classic data set for people's research. Later researchers focused on details and did a lot of work on feature extraction and classifier construction methods. Yiming Yang analyzed and compared various feature extraction methods, including Mutual Information, Information Gain, Statistics and so on, and tested almost all text classification methods on Retuters 21578 and OHSUMED datasets in subsequent research, and compared the performance of each classifier. In 1995, Vipnik developed a Support Vector Machine Algorithm based on statistical idea. The basic idea is to construct the optimal classified high-dimensional hyperplane. It can solve both linear separability and linear inseparability. The process is to transform the linear inseparable low-dimensional sample space into high-dimensional linear separable space through non-linear mapping, so it has attracted wide attention in the field of machine learning. Following the advent of Support Vector Machine Algorithm, Yoav Freund and Robert E. Schapire proposed AdaBoost algorithm and verified the rationality of the framework of AdaBoost algorithm in theory and practice. Later researchers have proposed similar lifting algorithms, which have been used in the research of text categorization and achieved some achievements.

2.2 Research on Knowledge Management
At present, the theoretical overview of knowledge management has not yet appeared, because it has relevant definitions in many disciplines and fields. Knowledge management has been studied earlier in foreign countries, and many experts in different fields have done research on it. For example, Drucker and Strassmann emphasize the growing importance of knowledge and information as resources. In the late 1970s, Everett Rogers from Stanford University and Thomas Allen from MIT made people realize how knowledge is generated, used and spread within an organization. With the development of Internet technology and computer technology, people have found a solution to knowledge management, and the theory of knowledge management has been put into practice.

The research of knowledge management system is just for practicing knowledge management. As early as 1978, an application named Augment, which was developed by Doug Engelbart, can provide interfaces for other systems and applications. In the 1980s, knowledge management system based on artificial intelligence and expert system appeared, and the concept of knowledge acquisition and knowledge system was put forward. At present, the products of knowledge management system in China
are becoming mature, and the HOLA system for enterprise content management developed by Deya Technology has been successfully applied to the Ministry of Commerce of the country; KMPRO knowledge management platform based on B/S framework developed by Deep Blue Sea Company can support a series of functions such as knowledge classification, browsing, association, Publishing and sharing, recommendation, etc. The above shows the maturity of enterprise-level knowledge management system products.

3. RELEVANT TECHNOLOGIES

3.1 Text Categorization Process
First, use the web crawler technology to write a crawler program for the specified category collection, crawl the corresponding knowledge text as the text training set of the category, and then construct the corpus collection, that is, the training set, and then the preprocessing of the training set, mainly The Chinese word segmentation technique is used to process the training set by word segmentation[2], and the stop words are removed. The dictionary is de-duplicated, and the pre-processed training set is extracted by the feature extraction method. Then the text representation method is used to process each. The document representation of the document. Preprocessing, feature extraction, and text representation are collectively referred to as feature engineering in the field of text classification. Finally, the text classification algorithm is used to construct the classifier to classify the text to be classified.

3.2 Model

3.2.1 Vector Space Model
Vector space model[3] was proposed by Salton of Cornell University in 1970 and applied to the famous SMART text retrieval system. Vector space model transforms texts into vectors in vector space, and the processing of texts becomes the operation of vectors in vector space, and the similarity of texts is expressed by the similarity of space vectors. In VSM, there are a series of terms describing the representation of text. Features refer to the basic language units that appear in the text and can represent the content of the text, mainly consisting of words or phrases. Documents generally refer to text and consist of feature items. In short, the word bag model is based on the dictionary that has been created, one by one to correspond to the number of corresponding words in each document, then you can get a vector of the same dimension and dictionary size. At this time, the text will be expressed as a vector, then the vector of the document can replace the document to do related calculations, such as similarity calculation, and the similarity between documents is text. Vector space model can describe the characteristics of documents because it is an important measure of this classification. However, because the vector space model only focuses on the number of occurrences of words, that is, word frequency, without considering the semantic relationship between words, the use of simple vector space model[4] is not very good for classification results. It needs to be processed in combination with other methods, which will be mentioned one by one below.

3.2.2 word 2vec model
Word2vec[5-8] mainly uses CBOW (Continuous Bag-of-Words Model) and Skip-Gram (Continuous Skip-Gram Model) models. Both CBOW model and Skip-Gram model are based on Huffman tree.
initial value of the intermediate vector stored by the non-leaf nodes in Huffman tree is zero vector, and the word vector corresponding to the leaf nodes is randomly initialized. The goal of CBOW is to predict the probability of the current word according to its context, whereas Skip-Gram, on the contrary, predicts the probability of the context based on the current word, as shown in Figure 1. Both methods use artificial neural networks as their classification algorithms. At first, each word is a random N-dimensional vector. After training, the optimal vector of each word is obtained by CBOW or Skip-Gram method.

(1) Model structure of CBOW

The model structure of CBOW is shown in Fig. 2, in which CBOW consists of three layers. The first layer is called the input layer, and the input is the word vectors of several words. The number of words is related to the random window size c (c denotes the number of current words covering forward or backward); the middle layer is called the mapping layer, and the input is the sum of several word vectors; the third layer is the output layer, and the output is the two words in the box. A fork tree is a Huffman tree constructed by weighting the number of occurrences of words in the corpus. All non-leaf nodes of the Huffman tree are associated with the nodes of the mapping layer. All leaf nodes represent all words in the corpus (assuming the number of words in the corpus is v), each leaf node corresponds to a word vector, and each non-leaf node is also a vector, but does not represent a word, but represents a category of words. In fact, the input word vectors are the same as some leaf nodes in Huffman tree. Of course, the input words and the final output words are not necessarily the same words (basically not the same word), but these words and the output words often have a semantic relationship. The function of CBOW network structure is to calculate the probability of target words in the current network structure through a given context.

(2) Model structure of Skip-Gram

The model structure of Skip-Gram is shown in Figure 3, where Wi is a previously computed word and Wi is directly connected to Huffman tree. This CBOW model structure has less mapping layer.

Word2vec model trains CBOW and Skip-Gram models on a given corpus, and then outputs the vector representation of all words that appear on the corpus. Based on the obtained word vector, we can...
calculate the relationship between words, such as word similarity, semantic relevance and so on. Considering the choice of corpus, the Chinese Wikipedia corpus is used to train the word2vec model, and the word vectors of each word are finally obtained.

### 3.3 TF-IDF

TF (Term Frequency) denotes the frequency at which a given word t appears in a document d. The higher the TF, the more important the word t is to document d. The lower the TF, the less important the word t is to document d. In practical documents, there are often some high-frequency words, symbols, punctuation or random codes which have little significance for text content recognition, such as "that, this, yes, in" and so on. If TF is directly used as the criterion of similarity evaluation, obviously there are problems. IDF (Reverse File Frequency) is the basis for judging the importance of a word to the entire document set or corpus. In a document set or corpus, the smaller the number of documents containing the word t, the larger the IDF, indicating that the word t has a good recognition and discrimination ability at the level of the whole document set. Therefore, the weighted word frequency calculation method based on TF-IDF is a statistical method. Its criterion is to judge the importance of words according to the number of times they appear in documents. If a word or short sentence appears frequently in one article (TF is high) and seldom in other articles (IDF is high), it is considered that the word or short sentence has good distinguishing ability and is suitable for classification. Therefore, TF describes the importance of a word t to a document, while IDF describes the importance of a word t to the entire document set[6].

\[
TF_t = \frac{n}{N}
\]

Among them, \( n \) denotes the number of occurrences of the word t in the document, and \( N \) denotes the sum of occurrences of all words in the document. The ratio of the two is the word frequency of the word. For example, "Linux" appears 10 times in the document, and the total number of occurrences of all words in the document is 500 times. Then the word frequency of "Linux" is 10/500 = 0.02.

For the IDF of a particular word, the total number of files divided by the number of files which contain the word and logarithm to this quotient obtained. Which can be expressed as:

\[
IDF_t = \log \left( \frac{|D|}{|\{t \in d \}|} \right)
\]

Where \( |D| \) is the total number of documents in the corpus and denominator is the total number of documents containing the word ti. For example, there are 100 documents, including 20 files containing the word "test", and the IDF of the word "test" is \( \log (100/20) = 1.609 \).

### 4. CLASSIFIER DESIGN

The design of classifier in this paper includes two stages: the construction of classification data set and the construction of classifier. Explanations and methods have been given in the previous chapters for the construction of categorized data sets and a corpus is constructed by using web crawlers to capture text data. The construction methods of text classifier are mainly based on pattern classification, which can be roughly divided into three categories: statistical-based methods, such as KNN classifier, Bayesian classifier, support vector machine, regression model, rule-based methods, such as decision tree, random forest, Association rule, and connection-based methods, such as artificial neural network. This paper adopts KNN algorithm and Naive Bayesian algorithm.

#### 4.1 Improved KNN Classifier Design and Performance Evaluation

#### 4.1.1 Classifier Algorithmic Steps

KNN classification algorithm is a classic, simple and effective classification method. It is easy to implement and has good classification effect. It has been used in the field of text classification for a long time.
KNN is classified by measuring the distance between different eigenvalues. The idea is to find the K most similar samples of a sample in the feature space. When most of the K samples belong to a certain category, the sample is also considered to belong to this category. The K value is usually an integer less than 20. In KNN algorithm, the selected adjacent samples are all correctly classified samples. In classification decision-making, the method only determines the category of the sample to be divided according to the category of one or more nearest samples.

In this paper, TF-IDF algorithm is used to calculate the TF and IDF values of words in each document, and the top 20 largest words of TF*IDF value are taken as the feature words of the document. Word2vec model is used to get the word vector of the feature words. Then each document has a feature matrix, and the similarity matrix is obtained through one-by-one comparison, and then the document similarity is obtained. The specific steps are as follows:

1. Assuming that a document D has been preprocessed, the TF*IDF value of each word is obtained according to the TF-IDF algorithm, and the first 20 words of the TF*IDF value are taken as the feature words of the document.

\[d = \{w_1, w_2, \ldots, w_{20}\}\]

2. Word 2vec model is used to transform \(w_i\) in D into vectors, and the word vector matrix D of feature words is obtained.

3. Document t to be categorized performs the above two steps to obtain the word vector matrix T of document t to be categorized.

4. At this point, all documents in the document set are transformed into word vector matrices.

5. The word vector matrices of documents to be classified are computed in order of similarity with the word vector matrices of each document in the document set.

6. The similarity matrix \(P\) is obtained from the fifth step.

\[
P = \begin{bmatrix}
  s_{11} & \cdots & s_{1n} \\
  \vdots & \ddots & \vdots \\
  s_{k1} & \cdots & s_{kn}
\end{bmatrix}
\]

Where \(s_{kn}\) represents the similarity value of the K word in document D and the N word in document t.

7. The average similarities \(Q_1\) and \(Q_2\) of document D and document T are obtained by the following formulas

\[
\arg\max_{Q_2} = \frac{1}{m} \sum_{i=1}^{m} \max\{\text{sim}(w_k, w_{i}), \text{sim}(w_k, w_{i}), \ldots, \text{sim}(w_k, w_{i})\}
\]

8. The arithmetic average of the maximum similarity of two-way average is obtained by using the following formula as the calculation value of document similarity.

\[
\text{sim}(d, t) = \frac{1}{2}(Q_1 + Q_2)
\]

9. Finally, according to the similarity between all documents in the document set and the documents to be classified, the first K maximum values are extracted to determine which category is the most, and then the categories of documents to be classified are determined.

4.1.2 Performance Evaluation
The categories and numbers of document sets used to verify the performance of KNN classifiers are as follows

| Document Category          | Document Number |
|----------------------------|-----------------|
| Artificial Intelligence    | 2000            |
| Software Testing           | 2000            |
| Linux Operating System     | 2000            |
| Interface class            | 2000            |
In this paper, the accuracy of prediction is used as an evaluation index of classifier performance. 85% of document sets are used as training sets, and 15% of document sets are used as test sets to test the accuracy under different K values.

![Classification performance](image)

Figure 4. KNN classification performance.

From the experimental results, we can see that different K values will affect the classification index, not that the bigger K values, the better the classification effect. From the graph, we can see that when K is greater than 20, it will almost continue to decline. So according to the experimental results, when K = 10, the correct rate reaches 77.6%. Therefore, this paper chooses K = 10 as the parameter of KNN classifier.

But K is not the only factor affecting the performance of KNN classifier. Secondly, there is the size of the document set. The experiment chooses a document set of 500-4000 for each category, increasing every 500 to verify the performance of the classifier in different document sets. The experimental results are shown in the following figure.

![Classification performance](image)

Figure 5. KNN classification performance.

From the experimental results, when the training set of each category is less than 2500, the classification effect is gradually improved, but beyond 2500, the performance of the classifier fluctuates little. Therefore, KNN classifier is not the more samples, the better the classification effect. When a certain threshold is reached, the classification performance tends to be stable.

4.2 Improved Naive Bayesian Classifier Design and Performance Evaluation

4.2.1 Classifier Algorithmic Steps
Naive Bayesian classifier\(^{(9)}\) algorithm belongs to Bayesian learning method which is efficient and widely used. When Naive Bayesian algorithm classifies text, it assumes that all document features are independent of each other. Assuming that \(d_i\) is any document in a given document set and belongs to a category \(c_i\) in document category set \(C = \{c_1, c_2, ..., c_n\}\), the theoretical basis of Naive Bayesian algorithm for calculating the category of a document class is as follows.

(1) Conditional probability: The core part of Naive Bayesian is the Bayesian rule, and the core of Naive Bayesian rule is conditional probability. The formula is as follows.

\[
P(c_i | x_1, x_2, ..., x_n) = \frac{P(c_i) \cdot P(x_1, x_2, ..., x_n | c_i)}{P(x_1, x_2, ..., x_n)}
\]

Where \(c_i\) stands for category, \((x_1, x_2, ..., x_n)\) represents feature attributes.

(2) Lexical Set Model: For a given document, only a word is counted whether it appears in this document or not.

(3) Word bag model: For a given document, statistics of the frequency of a word appearing in this article, in addition, often need to eliminate the very low importance of high-frequency words and stop words. Therefore, the word bag model is more refined and effective.

Based on the naive Bayesian theory, the difficulty now lies in how to calculate \(P(x_1, x_2, ..., x_n | c_i)\). In this paper, we need a dictionary to vectorize the document based on the word bag model. As mentioned earlier, after the corpus is created, stop words and high frequency words are taken out, and a dictionary is created, which includes all the necessary words in the training text set. Secondly, each processed document is vectorized according to the dictionary. Specifically, each document is defined as a dictionary size, which traverses each word in a certain type of document and counts the occurrence times. Finally, we get vectors of the same size as a dictionary, which are composed of integers. Each integer represents the occurrence frequency of a word in a corresponding position in the dictionary in the current document. Finally, the total number of vocabulary in each kind of document processed, the sum of the word frequency vectors of a certain kind of document divided by the total number of vocabulary in the corresponding category are counted. That is to say, the corresponding conditional probability is obtained, that is, the conditional probability of each feature attribute.

After the same vectorization of document \(d\), the probability of the occurrence of a category under the condition of current document \(D\) can be calculated according to the conditional probability obtained above, that is, comparing following three formulas.

\[
P_1 = P(c_1 | d(x_1, x_2, ..., x_n))
\]
\[
P_2 = P(c_2 | d(x_1, x_2, ..., x_n))
\]
\[
P_n = P(c_n | d(x_1, x_2, ..., x_n))
\]

\(P_1, P_2, ..., P_n\) is the probability size, the most probable category is the predicted category of documents to be classified.

4.2.2 Performance Evaluation

The categories and numbers of document sets used to verify the performance of Naive Bayesian classifiers are as follows

| Document Category       | Document Number |
|-------------------------|-----------------|
| Artificial Intelligence | 2000            |
| Software Testing        | 2000            |
| Linux Operating System  | 2000            |
| Interface class         | 2000            |

In this paper, the accuracy of prediction is used as an evaluation index of classifier performance. 85% of document sets are used as training sets and 15% of document sets are used as testing sets. The number
of training sets in this experiment increases from 500 to 5000 in turn and every 500 in turn. The experimental results are as follows:

From the graph, we can see that when the document set is 500 to 3000, the classification accuracy is in an upward trend. When the classification accuracy reaches 3500, the increase of the classifier accuracy is slight, but the overall maximum classification accuracy is about 72%. Analyzing the reasons, we find that when considering the feature attributes (where the feature attributes are the words in the document), we use the dictionary to classify each document. Based on the fact that the number of occurrences of a word in a document is regarded as a feature and the importance of a word in a document is neglected, the inverse document frequency (IDF) is introduced to measure the importance of a word in a document. For example, the frequency of occurrence of a word in its own class is very high and that of occurrence in other categories is very low. This word plays a key role in distinguishing this category of documents, i.e. the high frequency of inverse documents, which should be given higher weight in calculation.

Based on the improved classification algorithm, we introduce IDF to Naive Bayesian algorithm to repeat the above experiments. The results are as follows:

From the result chart, we can see that after introducing IDF, the classification accuracy is improved as a whole, and the highest classification accuracy rate is 78.9%.
Comparing the experimental results before improvement with those after improvement, as shown in the figure above, we can find that when the document set is less than 2500, the correct rate increases as a whole, but the effect is not obvious. When the document set is more than 3000, the correct rate before and after improvement is quite different. The introduction of IDF plays an important role in the optimization of classifier performance.

5. SUMMARY
By designing the algorithm of KNN classifier and Naive Bayesian classifier, and evaluating the performance index of each classifier through experiments, it is found that KNN classifier can achieve better classification effect only when the number of document sets of various documents is equal. If the number is not appropriate, it will cause imbalance of training data, then the imbalance of data for KNN classifier is good. The parameter k of the classifier has a great impact on the overall classification effect of the classifier, so when using KNN classifier, try to ensure the balance of data. For naive Bayesian classifier, the imbalance of data sets has little effect on the performance of classifier, because naive Bayesian classifier is based on naive Bayesian rule, and the core of naive Bayesian rule is based on conditional probability, so when using naive Bayesian classifier, we do not need to consider the balance of data quantity. Therefore, classifier should be rooted. According to the situation of text training set, we use KNN classifier and Naive Bayesian classifier in the design of knowledge management system. We can choose the appropriate classifier according to the specific data set and apply it to knowledge management system to increase the classification accuracy of knowledge management system and improve its function.

REFERENCES
[1] Wen,Q. Text Categorization based on Text Feature Extraction[J]. Computer Knowledge and Technology, 2018,14(18):188-189+192
[2] Li,J,L,Wang,P,Q,Han,Q,Y Research on Chinese Word Segmentation Technology for Military Field[J]. Computer and Modernization. 2018(11):115-118+126
[3] Wibowo, A., Handoyo, A., Halim, A.. Application of Topic Based Vector Space Model with WordNet[P]. Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on,2011
[4] Lu,X. Research of Support Vector Machine Multi-classification Model Basedon Granular Computing[D]. North China University of Science and Technology.2015
[5] Jixian Zhang,Qinglin Wang,Yuan Li,Dongmei Li,Yuexing Hao. A Method for Chinese Text Classification Based on Three-Dimensional Vector Space Model[P]. Computer Science & Service System (CSSS), 2012 International Conference on,2012.
[6] Nakanishi, T.. Semantic Context-Dependent Weighting for Vector Space Model[P]. Semantic Computing (ICSC), 2014 IEEE International Conference on, 2014
[7] Dongwen Zhang, Hua Xu, Zengcai Su, Yunfeng Xu. Chinese comments sentiment classification based on word2vec and SVM perf[J]. Expert Systems With Applications, 2015, 42(4)
[8] Wang, J., Luo, L., Wang, D., Q. Research on Chinese Short Text Classification Based on Word2Vec[J]. Computer Systems & Applications. 2018, 27(05): 209-215
[9] Ding, Y., Wang, X. M. Naive Bayes classification algorithm based on feature weighting[J/OL]. Application Research of Computers