Classification of Medical Consultation Text Using Mobile Agent System Based on Naïve Bayes Classifier

Xingyu Chen[1,4(✉)], Guangping Zeng[1,4], Qingchuan Zhang[2,4], Liu Chen[1,4], and Zhuolin Wang[3]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China
cscserer@sina.com, zgp@ustb.edu.cn, chenliueve@163.com
[2] School of Computer and Information Engineering, Beijing Technology and Business University, Beijing, China
zqcl982@126.com
[3] School of Humanities and Social Science, University of Science and Technology Beijing, Beijing, China
zhuolinwang@yeah.net
[4] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, China

Abstract. Aiming at the interaction model of the Internet medical website, a classifier of medical text data based on Naive Bayes was proposed and realized in this paper. Once a user posed questions on the websites, this classifier would instantly classify the user's questions and enable accurate question delivery. Furthermore, a data service platform was realized by taking advantages of mobile agent technology. With the service platform, companies could avoid considering the security of data when conducting data analysis. Finally, experiments were conducted according to the process of data analysis in the service platform. The experimental results showed: the proposed service platform was feasible, and a medical consultation text classifier with high accuracy was realized to improve user experience of medical websites.

Keywords: Naive Bayes · Medical big data · Mobile agent
Artificial intelligence

1 Introduction

With the rapid development of the internet and the great enhancement of people's awareness of health, the internet is becoming an important channel to acquire medical information. By posing questions on medical websites, persons can get answers by professional doctors. After users having putting forward their questions, there would be doctors online to browse those questions. They will answer questions belonging to their special field and give their suggestions. Because most medical websites adopt a Q&A (Question and Answer) system with blackboard mechanism, the time effectiveness is poor. Users have to wait for doctors to answer them, while doctors have to spend time

K. Long et al. (Eds.): 5GWN 2017, LNICST 211, pp. 371–384, 2018.
https://doi.org/10.1007/978-3-319-72823-0_35

browsing questions and deciding which and what to answer. In order to improve the experience for both users and doctors and increase the effectiveness of the system, data mining methods based on medical big data can be taken to analyze question texts [1]. With these methods, questions would be pushed prior to the doctors who are most likely to answer. Besides, users would be offered with reference materials before getting replied.

Medical big data plays an important role in the field of big data [2]. With the popularity of the mobile medical, internet medical, automatic analysis detectors, wearable devices, etc., all parties including patients, doctors, companies and the environments are becoming direct creators of data, generating mass medical data every day.

Compared with big data of other fields, medical big data have almost covered all the personal information of citizens, from the most private information of body and disease to the information of personal property, accommodation, medical insurance and so on. Therefore, when using medical big data, technical personnel should not only consider about security requirements from companies, hospitals and other providers, but also consider about the privacy of the data. Generally, there are two ways of conducting big data analysis: the first is that the companies which hold data provide technical supporters with data to carry out data analysis; the second is that the technical supporters appoint personnel to companies. In the first way, technical supporters have to insure the data security. In the second way, the results of data analysis are hard for promotion to create greater value. Because of the disadvantages of the ways mentioned above, what way should be taken to conduct data analysis has become a great concern of the companies and the academia. A reasonable way can not only reduce the cost of the enterprise, but also make the new technology produce more value.

In this paper, researchers have realized a text classifier based on Naive Bayes model with higher accuracy aiming at the process of the Q&A [3–5]. The classifier can help to quickly classify problem descriptions to different departments and to pre-diagnose the problems. Furthermore, the authors have designed a data analysis process based on mobile agent technology where a mobile agent data service platform has been realized [6, 7]. This service platform can make use of the mobility and self-determination characteristics of mobile agent. With this platform, the problems concerning the data security would be solved to some degree. What's more, the new data analysis technology would be used by more companies and clients.

2 Paper Preparation

2.1 Naïve Bayes

Naive Bayes classifier model is a kind of simple probability classifier applied in the independence assumption Bayes theorem. It assumes that each features are not related, depends on accurate natural probability models and enable to get very good classification effect in supervised learning sample sets. The classification process shows as follows:

(1) Using a dimensional feature vector $X = \{x_1, x_2, \cdots, x_n\}$ to represent each data sample, which separately describes n features $A_1, A_2, \cdots, A_n$ of samples.

(2) Assuming there are m classes $C_1, C_2, \cdots, C_n$. Given an unknown data sample X, the classification would predict that X belongs to the class with the Maximum a Posteriori (MAP) under the condition X. In other words, Naive Bayes classification would allocate the unknown sample X to class $C_i (1 \leq i \leq m)$, if and only if

$$P(C_i|X) > P\big(C_j|X\big),\ j = 1, 2, \cdots, m,\ j \neq i \tag{1}$$

Class C_i which would enable the Maximum a Posteriori $P(C_i|X)$ is called the Maximum a Posteriori Assumption. According to Bayes' theorem,

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{2}$$

(3) Because $P(X)$ is invariant, we just need to ensure the maximum $P(X|C_i)P(C_i)$. If the prior probability of class C_i is unknown, we generally assume that these classes are equiprobable, which is $P(C_1) = P(C_2) = \cdots = P(C_m)$. Therefore, the problem is converted to maximize $P(X|C_i)$. Otherwise, the prior probability of class C_i would be calculated by $P(C_i) = s_i/S$, where s_i is the number of training samples in class C_i while S is the total number of training samples.

(4) Given data sets with many features, we might cost too much to calculate $P(X|C_i)$. In order to lower the cost, we could assume that each features of samples are mutually of conditional independence, which means there is no dependency among each features, then

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{3}$$

The probability $P(X|C_i)$ can be estimated by training samples.

(5) This means that under the above independence assumptions, the conditional distribution over the class variable C is

$$P(C_i|X) = P(X|C_i)P(C_i) = P(C_i)\prod_{k=1}^{n} P(x_k|C_i) \tag{4}$$

And a Bayes classifier, is the function that assigns a class label $\hat{y} = C_i$ for some i as follow

$$\hat{y} = \underset{i\in\{1,\ldots,m\}}{\operatorname{argmax}}\, P(C_i)\prod_{k=1}^{n} P(x_k|C_i) = \underset{y}{\operatorname{argmax}}\, P(y)\prod_{k=1}^{n} P(x_k|y) \tag{5}$$

A class's prior may be calculated by assuming equiprobable classes, or by calculating an estimate for the class probability from the training set. To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set [5]. The assumptions on distributions of features are called the event model of the Naive Bayes classifier.

2.2 Multinomial Event Model

With a multinomial event model [8], the distribution is parameterized by vectors multinomial $p_y = \{p_{y_1}, p_{y_2}, \cdots, p_{y_n}\}$ for each class y, where n is the number of features (in text classification, the size of the vocabulary) and p_{y_i} is the probability $P(x_i|y)$ of feature i appearing in a sample belonging to class y.

The parameters p_y is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{p}_{y_i} = \frac{f_{y_i} + \alpha}{f_y + n\alpha} \tag{6}$$

Where f_{y_i} is the eigenvalue of x_i, and f_y is the total count of all eigenvalue for class y.

The smoothing priors $\alpha \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations.

2.3 TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus [9]. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times that a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

Term Frequency is the number of times that a term w occurs in a document d. If we donate the number of times by $count(w, d)$ and the total number of word occurs in d by $size(d)$, then

$$TF(w, d) = \frac{count(w, d)}{size(d)} \tag{7}$$

The inverse document frequency is a measure of how much information the word provides, that is, whether the term is common or rare across all documents.

$$IDF(w) = log(\frac{n}{docs(w, D) + 1}) \tag{8}$$

$docs(w, D)$ is the count of documents that contain the word w. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $docs(w, D) + 1$.

Then $TFIDF(w, d)$ is calculated as

$$TFIDF(w, d) = TF(w, d) * IDF(w) \tag{9}$$

3 A Medical Consultation Text Classifier Based on Naive Bayes

With the rapid development on the Internet and the great enhancement on health awareness, using fence netting is becoming an important way to acquire medical information. People can put forward questions on medical websites and get answered by professional doctors. Their questions are generally posed as describing a series of symptoms to get disease diagnosed or seeking for notes and directions. For example,

(1) The cold is very afflictive wow! Rhinitis how should do?
(2) Darling 14 months, cold, have a fever, snorty, sleep to still be met shy, whats do not eat, how to do?

Those questions would be checked and answered by professional doctors sooner or later.

Through training analysis on the history Q&A data of medical websites, we have developed a classification method on medical consultation text – a smart consulting and diagnosing classifier. Once a user has submitted a question, this applied classifier would immediately deduce what possible disease the user wants to consult and which department he should turn to. That's to say, this classifier can not only quickly provide users with resources of related diseases, but also accurately recommend questions to experts in related fields.

3.1 Data Analysis

The hierarchical structure diagram of original data is shown as Fig. 1.

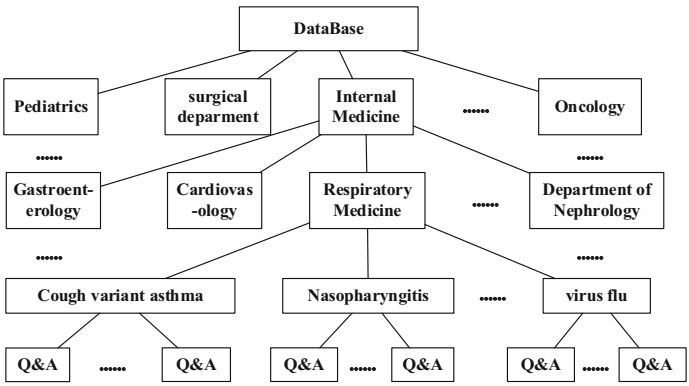

Fig. 1. The first layer is the total database. Each node of the second layer represents a comprehensive department. Each node of the third layer represents a specific department. Each node of the fourth floor represents a specific disease. Each node of the fifth floor represents a specific set of (Q&A).

According to requirements, we haven't carried out data analysis on comprehensive departments. It means that the nodes of the second layer have been excluded, and the nodes of specific departments have directly connected to the root node.

Classifying users' questions can be separated into two steps. The first step is to classify the departments corresponding to the questions. The second step is to classify the diseases under the charge of the departments. In this way, the training of the first-step classifier needs all the Q&A data, while the training of the second-step classifier just needs the Q&A data of a specific department. Therefore, the data sets used by the second-step classifier are the subsets of the first-step classifier. We have taken a data set of a specific department as the sample data. By analyzing the data set, we have implemented the function module which can figure out and generate classifier.

3.2 Data Pre-processing

In order to develop a dependable classifier, we have conducted data pre-processing during which we have acquired the terminology library and set the stop words library. Then we have conducted Chinese word segmentation and feature abstraction of text keywords. The terminology library refers to all the words and terms related to the medical field, which are the basic medical knowledge, including names of diseases, symptoms, body parts, medicines, and so on.

Setting up the terminology library makes contributions to better conducting Chinese word segmentation. Nouns of this part can be obtained from websites such as Medical Encyclopedia and Baidu Encyclopedia. Having considering that all the data of hospital departments are needed to design the processes of training classifiers, we have used web-crawler to access all the names of symptoms and diseases.

Setting up the appropriate stop words library can improve the accuracy of classifier. The specific operation is to identify the stop words library during feature abstraction of text keywords. In this way, the words belonging to the stop words library would not be included in the list of keywords. The stop words library contains words of high occurrence frequencies but of no prior supports for classification. Through sample data analysis, we conclude that some words with special parts of speech are useless in classification, such as pronouns, conjunctions, idioms, punctuation marks. If we identify these words as the stop words, the accuracy of our classification would be greatly improved.

The following pseudo-code describes how to access the appropriate stop words library before data training.

The algorithm of dynamic access to the appropriate stop words library. Input: $D = \{d_1, d_2, \cdots, d_n\}$, referring to the set of question text data; $stopProperty = \{p_1, p_2, \cdots, p_k\}$, referring to the set of parts of speech of stop words; *Dictionary*, referring to the terminology library used in Chinese word segmentation. Output: $stopWords = \{w_1, w_2, \cdots, w_m\}$, referring to the stop words library applied to D.

```
begin
  D = readTrainData()
  stopProperty = readStopProperty()
  Tool.load(Dictionary)
  stopWords = emptyList()
  wordPropertyListD = Tool.cut(D)
  foreach doc in wordPropertyListD
    foreach (word, prop) in doc
      if prop in stopProperty then
        stopWords.append(word)
  return stopWords
end
```

3.3 The Training Process of the Medical Consultation Text Classifier

The training algorithm process of the classifier based on Naive Bayes is as follows:

(1) Read out the Q&A data from database according to the classification.
(2) Upload the terminology library and the stop words library;
(3) Carry out word segmentation to all the question data. Because Q&A data are all in Chinese, Chinese word segmentation is needed;
(4) Calculate the TF-IDF eigenvector of the training data, and use an eigenvector to represent a question;
(5) Divide data into training data sets and testing data sets;
(6) Use multinomial Naive Bayes classifier model to get data training, and then get a multinomial Naive Bayes classifier based on TF-IDF;
(7) Use testing sets to test the accessed classifier.

This algorithm is described as the following pseudo-code:

The training algorithm of the classifier based on Naive Bayes. Input: $D = \{(d_1, t_1), (d_2, t_2), \cdots, (d_n, t_n)\}$, referring to the set of question text data with class identifiers; $stopWords = \{w_1, w_2, \cdots, w_m\}$, referring to the stop words library; *Dictionary*, referring to the terminology library used in Chinese word segmentation. Output: *CLF*, referring to the classifier; precision, referring to the accuracy of its tests.

```
begin
  D = readData ()
  stopWords = readStopWords()
  Tool.load(Dictionary)
  docs = Tool. chineseSplit(D)
  vec = vectorizer(docs, stopWords)
  (dataTrain, dataTest) = dataDivide(vec)
  CLF = MultinomialNB(dataTrain)
  precision = Test(CLF, dataTest)
  return CLF, precision
end
```

4 The Design of Medical Big Data Service Platform Based on Agent

With the application of mobile agent technology, the design of big data service platforms can better solve the security and privacy problems of data. A mobile agent is a program substituting for people or other program to perform certain tasks. It can move from a mobile agent environment (MAE) to another in the complex and heterogeneous network system. It can choose when and where to move to search for appropriate resources. It can be suspended according to requirements, and then restart or continue to execute. It also can take the advantage of being in the same host as the resources – processing or using these resources nearby, accomplishing specific tasks and returning results and messages in the end.

Mobility and autonomy are the two important characteristics of mobile agent. These two characteristics can be used to design a new solution which makes efficient use of distributed resources and the network.

A mobile agent system is made up of mobile agent and mobile agent service environment (the mobile agent platform). A common agent includes the security service module, environment interaction module, function library, internal state set, the routing policy, constraint condition and the task solving module, and these structures are mutually related. Generally, a mobile agent carries tasks, while the task solving module finally executes these tasks. During solving process, the task solving module should satisfy the constraint condition assigned by the builder.

4.1 The System Structure of the Medical Big Data Service Platform

The service platform in this paper is designed based on traditional mobile agent platform, and its system structure is shown as the following Fig. 2.

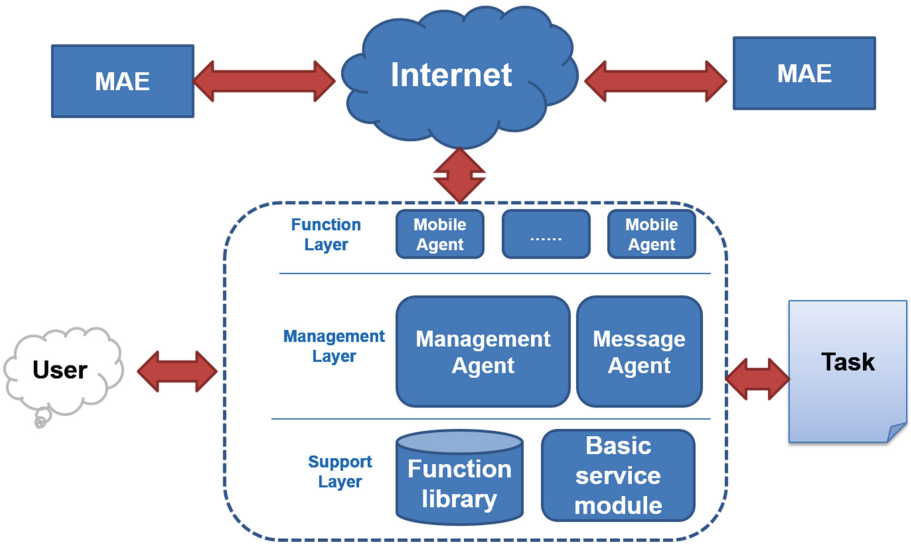

Fig. 2. The system structure of an agent platform is mainly made up of the following parts: message agent, management agent, task agent, function library and basic service module.

Message agent: in charge of receiving and sending messages between the platform and the outside, including: ① interacting with clients; ② receiving and sending messages of other agent platforms and task agent. The massage agent is only responsible for the interaction with the outside, while the management agent is responsible for understanding and processing messages.

Management agent: in charge of decision-making of the platform, including: ① setting up task agent by visiting function library; ② scheduling task agent, including executing and dispatching; ③ understanding and processing mutual information of message agent; ④ informing message agent to send messages or task agent to the outside.

Task agent: in charge of conducting specific tasks.

Function library: in charge of ① forming function modules; ② storing achievable and specific functions of task agent for management agent to schedule. The management agent can combine several function modules as a task solving module of the task agent.

Basic service module: in charge of providing necessary and basic service including directory services and security services.

The system structure of Client's agent platform and the data service platform are basically the same, but their difference lies in whether the function libraries of master station provide more function or not.

4.2 The Working Process of the Medical Big Data Service Platform

The working process is shown as the following Fig. 3.

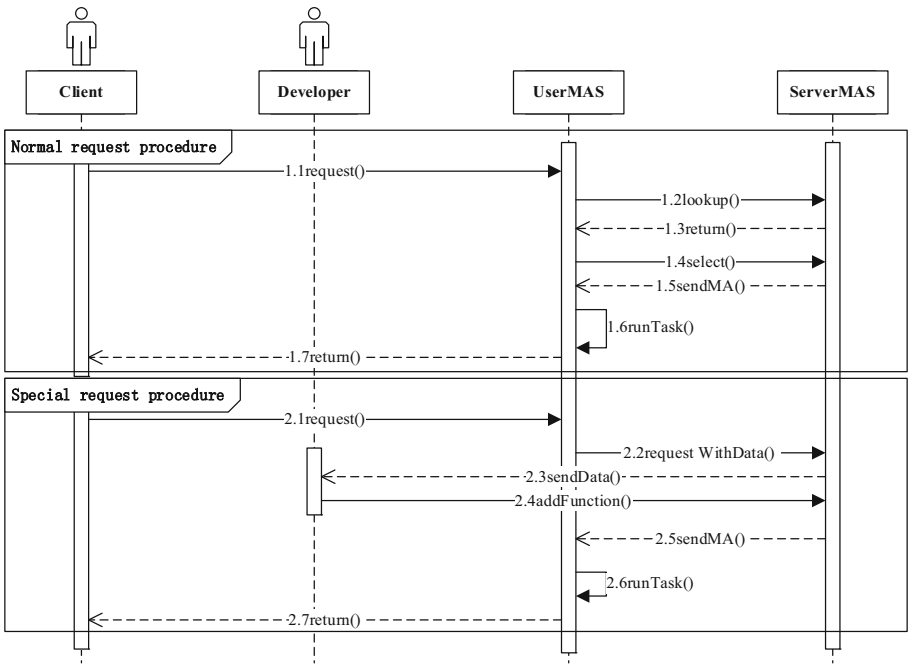

Fig. 3. Sequence diagram of working process on the medical big data service platform.

When Client need to take data analysis, he can conduct normal request procedure: Choose an existing function module of the service platform via his agent platform. The service platform only needs to put this function module into a task agent and then remove the task agent to the client's mobile agent platform. At last, practical data analysis and work would be accomplished.

If there is no suitable function module in the service platform, or if the client asks for special designing, technical personnel would write new function models according to the requirements of the client, as long as sample data and knowledge of the problem domain are given.

After accomplishing task solving, the task agent would return to service platform with results or execute operations such as instant death. Operations are determined by customer requirements and the specific situation.

5 The Experiment

5.1 Experimental Design

During the experiment, two computers with Linux system have been used to conduct the experiment. Consult texts provided by a large-scale Chinese medical consultation website have been taken as experimental examples. According to the provided data, the amount of texts reached 64060, related to 44 departments and 4801 kinds of diseases. Among them, 3950 kinds of diseases of 42 departments have texts under the catalogue. The 1553 texts of respiratory medicine department have been taken as sample data.

The classification experiment of sample data was conducted in a PC applied with mobile agent. With a multinomial Naive Bayes classifier based on TF-IDF, the conductors have taken the 1553 texts of respiratory medicine department as the data of the tuning model. During the experiment, conductors adopted the way of 10-fold cross-validation to get the accuracy of classification. Considering that the order of the data may affect the result of the classification and that the cross-validation on data segmentation is random, the average value of experiments have been used as the final result.

Sample data were used for parameter tuning of the generative process of the classification model. This process would be encapsulated into a task agent after the experiment and parameter tuning. Then, another platform turned to the service platform for the task agent, used 64060 Q&A texts for training and tested them by 10-fold cross-validation. With the above, the average value of experiments represented the final result.

5.2 Comparative Experiment

In order to validate the dialectical ability of the algorithm we used in the data set provided by the enterprise, several mature classification algorithms were selected to participate in testing experiments in this section. Using the classification results of all the data, we have compares the classification performances [10–14].

Here listed are the algorithms involved in the experiment:

(1) Decision Tree Algorithm [15]
(2) Random Forest Algorithm [16]

(3) k-Nearest Neighbor (KNN) [17]
(4) Support Vector Machine (SVM) [18]
(5) Naive Bayes with Multinomial event model using TF feature (MultinomialNB_TF)
(6) Naive Bayes with Bernoulli event model using TF-IDF feature (BernoulliNB)
(7) Naive Bayes with Multinomial event model using TF-IDF feature (MultinomialNB_TFIDF)

KNN algorithm has a key parameter k value need to be selected. The k value is the empirical parameter, which indicates the number of the selected neighbours. The selection of its value has a significant effect on the classification performance. In the experiment, the optimal value of k is not determined. We conducted the experiment using 1 to 20 as the k value, finding that with the k value increasing, the classification performance decrease. So we choose 1 as the k value.

During the whole experiment, the author used the same Chinese word segmentation tools, terminology library and stop word library.

5.3 The Classification Result

We have conducted 10-fold cross-validation many times on the sample data, expecting to get the comparatively stable process of generating the classifier. The results of the 5 times classification of respiration medicine department are shown as the following Table 1.

Table 1. The results of the 5 times classification of respiration medicine department

	1	2	3	4	5	6	7	8	9	10	Mean
1	86.5	82.7	86.5	83.2	88.4	88.4	88.4	87.7	89.0	87.1	86.8
2	87.8	87.8	81.4	85.2	87.1	89.0	86.5	87.7	85.8	87.7	86.6
3	89.1	89.1	87.2	86.5	85.8	90.3	84.5	87.1	86.5	90.3	87.6
4	90.4	88.5	87.8	82.6	84.5	83.9	89.7	89.7	89.0	88.4	87.5
5	88.5	82.1	86.5	85.8	87.1	84.5	90.3	85.2	87.7	84.5	86.2

Final result: 86.9%.

The classification results of all the data are shown as the following Table 2.

Table 2. The classification results of all the data

	1	2	3	4	5	6	7	8	9	10	Mean
1	86.4	86.0	85.0	86.2	85.8	85.2	86.2	85.4	86.0	85.6	85.8
2	86.0	85.7	86.8	85.8	85.5	86.0	85.2	86.2	85.3	86.0	85.9
3	85.5	85.6	86.3	85.9	85.8	85.7	86.6	85.9	86.2	85.2	85.9
4	86.2	85.2	85.8	86.2	85.4	85.7	84.9	86.0	86.3	85.9	85.8
5	85.5	85.5	85.4	86.1	86.3	85.5	85.5	85.5	86.5	85.3	85.7

Final result: 85.8%.

The results of pre-diagnosing diseases under each department are shown as the following Fig. 4.

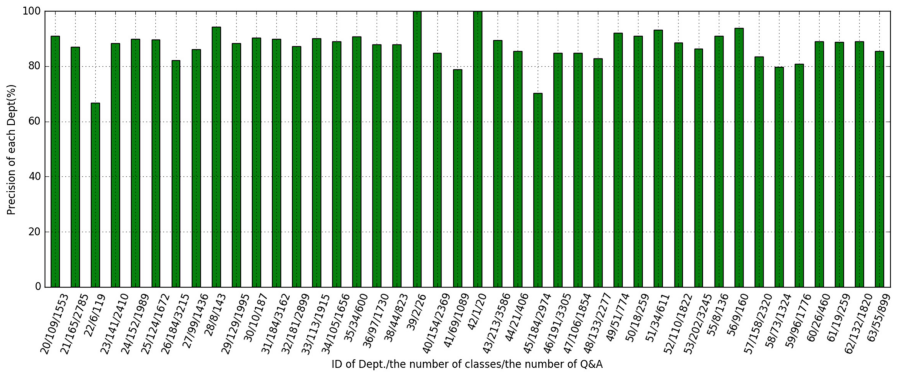

Fig. 4. The results of pre-diagnosing diseases under each department.

The experimental results of the comparative test are shown as following Fig. 5.

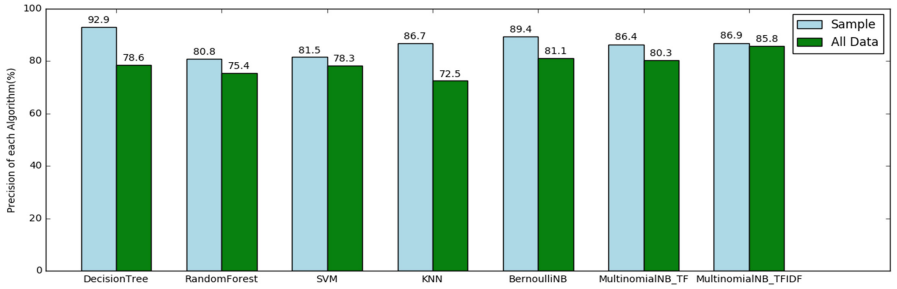

Fig. 5. The experimental results of the comparative test.

5.4 Experiment Analysis

The experiments above were carried out according to the data analysis process of the companies in the service platform. The results of the experiment can prove that the mobile agent data service platform proposed and implemented in this paper is feasible.

First of all, the data in Table 1 show that we have a classifier with the accuracy at 86.9% when experimenting with data from the Department of Respiratory Medicine. The data in Table 2 show that we can get a classifier with the accuracy at 85.8% for all data. These number means when the user publishes a new medical consultation question, the system will recommend the text to the doctors. There are average seventeen recommendations which are successful in twenty, other three recommendations need to be manually marked and re-pushed. And there are average fifteen people in

twenty will get the right information as soon as possible, other five will get help after the question is reposted. And eventually they will get the guidance of doctors.

Secondly, by using the method of classifier generation for text classification in each department, the average classification accuracy is 87.7%.

At the last, through the comparison experiment, we can figure out that the Naive Bayesian algorithm using TF-IFD characteristic polynomial in this paper has good accuracy in the general data set. But through the comparison of the sample data, the accuracy rate of Decision Tree is as high as 92.9%, which indicates that Decision Tree is more accurate than Naive Bayes algorithm in pre-diagnosing disease under specified department. In order to confirm the conclusion, this paper uses Decision Tree to classify diseases in all departments. Result shows the average accuracy rate of Decision Tree algorithm is 93.1%. With the conclusion, those two algorithms can be combined to pre-diagnosing disease to get better performances.

6 Conclusion

The research of this paper is focused on the current research hotspot – medical big data. First of all, for the purpose of optimizing the user experience on the interaction process of this kind of websites, researchers have made deep understanding on a famous large-scale medical consultation website, and have conducted data analysis with the historical data. Secondly, with the analysis results, a classifier of medical text data based on Naive Bayes was proposed and realized to find out valuable medical logic knowledge. In addition, in order to meet the requirement for data confidentiality when doing outsourcing data analysis and to maximize the values of technology or models from data analysis, this paper has carried on the discussion on the issue. Finally, a prototype systems of data analysis service platform has been designed and realized using mobile agent technology based on Naive Bayes medical text classifier.

Through experiment and analysis, we have validated that the classifier based on the Naive Bayes can realize better classification of medical consultation texts and stability compared with other algorithms. Moreover, Decision Tree can better pre-diagnose the questions. The two algorithms can be combined to pre-diagnosing disease to get better performances in the reality.

Acknowledgements. Our work was supported by National High-tech R&D Program (863 Program No. 2015AA015403).

References

1. Russel, S.J., Norvig, P.: Artificial Intelligence - A Modern Approach. Prentice Hall, Upper Saddle River (2003)
2. Jee, K., Kim, G.H.: Potentiality of big data in the medical sector: focus on how to reshape the healthcare system. Healthc. Inf. Res. **19**(2), 79–85 (2013)
3. Zhang, H.: The optimality of Naive Bayes. In: Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA (2005)

4. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam filtering with Naive Bayes - which Naive Bayes? In: CEAS 2006 - The Third Conference on Email and Anti-Spam, Mountain View, California, USA, 27–28 July 2006
5. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers, pp. 338–345 (2013)
6. Essa, Y.M., Attiya, G., El-Sayed, A.: New framework for improving big data analysis using mobile agent. Int. J. Adv. Comput. Sci. Appl. **5**(3), 25–32 (2014)
7. Gray, R.S., Cybenko, G.: Agent TCL: a flexible and secure mobile-agent system. In: Proceedings of the 1996 TCL/TK Workshop, pp. 9–23 (1999)
8. Jin, X., Zhou, W., Bie, R.: Multinomial event naive Bayesian modeling for SAGE data classification. Comput. Stat. **22**(1), 133–143 (2007)
9. Robertson, S.: Understanding inverse document frequency: on theoretical arguments for IDF. J. Doc. **60**(5), 503–520 (2004)
10. Baker, L.D., McCallum, A.K.: Distributional clustering of words for text classification. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 96–103 (2003)
11. Rogati, M., Yang, Y.: High-performing feature selection for text classification (2003)
12. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. **2**(1), 45–66 (2002)
13. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: AAAI 1998 Workshop on Learning for Text Categorization, vol. 62(2), pp. 41–48 (2001)
14. Nigam, K.: Using maximum entropy for text classification. In: IJCAI 1999 Workshop on Machine Learning for Information Filtering, pp. 61–67 (1999)
15. Saad, M.K., Ashour, W.: Arabic text classification using decision trees. In: International Workshop on Computer Science and Information Technologies, CSIT 2010 (2010)
16. Bouaziz, A., Dartigues-Pallez, C., da Costa Pereira, C., Precioso, F., Lloret, P.: Short text classification using semantic random forest. In: Bellatreche, L., Mohania, M.K. (eds.) DaWaK 2014. LNCS, vol. 8646, pp. 288–299. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10160-6_26
17. Han, E.-H., Karypis, G., Kumar, V.: Text categorization using weight adjusted *k*-nearest neighbor classification. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 53–65. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45357-1_9
18. Colas, F., Brazdil, P.: Comparison of SVM and some older classification algorithms in text classification tasks. In: Bramer, M. (ed.) Artificial Intelligence in Theory and Practice. IFIP AICT, vol. 217, pp. 169–178. Springer, Boston (2006). https://doi.org/10.1007/978-0-387-34747-9_18