Toward an Intelligent Agent System using Document Classification Techniques

Jui-His Fu[2] , Jui-Huns Chang[1], Yueh-Min Huang[1,*], Sing-Ling Lee[2], and Wei-Guang Teng[1]

[1]Department of Engineering Science, National Cheng Kung University
Tainan 701, Taiwan, ROC
{n9895115, *huang, wgteng}@mail.ncku.edu.tw

[2]Department of Information Engineering and Computer Science, National Chung Cheng University
Chia-Yi 621, Taiwan, ROC
{fjh93, singling}@cs.ccu.edu.tw

Received 13 December 2006; Revised 5 March and 26 March 2007 ; Accepted 30 March 2007

Abstract. Because the rapid development of science and technology leads to rapid changes of the social structure, this paper discusses how to make use of the technological power to change the current organization, principle, management and process, to enhance the chief productivity and to reduce the human resource burden and improve the efficiency. Through this research, an intelligent document agent system is established, which has the agent function and classification techniques; this system is also expected to be able to adopt the agent function to improve the system design. This system makes a labeling format related to the document through XML, and takes it as the data transmission media including the Chinese information exchanged on the website; the system also selects the information among the transmission and finds out the information of the keywords through Chinese Sentence Segment, and classifies trains and increases the accuracy of data classification. Meanwhile, this system provides the classification error mechanism to improve the accuracy. I hope this theory can be adopted in schools, government and through the system, the domestic institutions and business organizations can find an economical, practical and reliable format to shorten the time of the artificial classification and document transmission, as well as improving the efficiency and performance of document classification.

Keywords: document classification, intelligent agent system, XML, term extraction

1 Introduction

As the information is more and more developed and people pay more and more attention to use computers to improve work efficiently, the concept of making use of agent technology to reach these goals is more and more welcomed by the society. In order to simplify the trouble and inconvenience from the artificial flow, the government decides to set up a new relative system to simplify the internal management.

Generally, the communication between two agencies depends on the document information contact; however it always costs a lot of time to deal with the document transmission and artificial inspection and receives only a very low efficiency, therefore, the electronic document techniques are accepted by more and more people. To speed up the policy information transmission, the government establishes the electronic document exchange system, which mainly makes a labeling format related to the document through XML[1] as the transmission format.

When the electronic document is delivered to each agency through electronic document exchange system, the staff often classifies the document into unit or individual by his or her personal experience, so it often occurs that the document is delivered to wrong place and it takes a long time to return the document. To deal with this problem and seek for a high efficiency, the government holds an intelligent agent system to automatically classify the document, which can increase the transmission accuracy and avoid the experience of staff turnover. Meanwhile, this system can also draw the results of artificial distribution which is available in Chung Cheng University in Taiwan as the standard of distribution efficiency and test accuracy so as to adjust its own accuracy.

The agent system in this paper is able to fully cooperate with the original operating system like document flow and electronic document signing system, which is the very aim of establishing such intelligent document agent system.

[*] Correspondence author

This paper consists of four parts, which leads to processing, data storage and extraction, document processing agent including intelligent document classification agent and wrong document agent, and interface agent.

We used real documents collected at National Chung Cheng University in Taiwan as our testbed to evaluate and verify both the effectiveness and efficiency of our approach. An integrated agent system whose functionalities include data archives, data preprocessing, classification, and dispatching is to be developed in this paper. To illustrate corresponding issues with more details, an example document in XML format with pre-defined tags and structures.

As for the system, its web structure is in MVC format and it takes J2EE as the research base, because J2EE is of the cross-platform convenience and the fast deploying function; meanwhile, it can be enlarged in accordance with the demand. J2EE is able to settle down a group of standard functions and support the distributed software; at the same time it can simplify the creation process by using the Component-based Application Model and improve the creation efficiently. Therefore, we will design a module---a document agent module, so as to have a wider and more flexible application with other systems in the future.

This paper is organized as follows: Section 2 gives some literal reviews. Section 3 describes the structure of our proposed Intelligent Agent System. Section 4 concerns about the improvement of the proposed Intelligent Agent System Techniques in the document flow and how of this technique will be practiced in the future. Finally, in the Section 5, a brief conclusion is made.

2 Related Works

In this paper, we introduce how to use techniques of document classification to implement an intelligent agent for dispatching official documents easily and precisely. We will discuss some related methods commonly used in document classification

2.1 Document Classification Agent

The agent on the basis of software design is called software agent which has two main fundamental features, autonomy and continuity. At the necessary moment, the software agent can take action without artificial help and will continuously run because of its autonomy factor. [2]

In this knowledge-explosion era, a huge pool of information and data requires the system management, so the agent used in document classification is becoming more and more important.

Application of single agent

The agent, adopted on the internet to classify the related economic news [3], can automatically gather, filter, classify and deliver the information through the network [4]; as the problem of information overload and recall on the WWW, a PAW agent is able to observe the website visited by the users and provide other similar websites to the user [5]; Bayesian classifications can get rid of the heterogeneous distributed grid data [6];

Application of multi-agent

Compared with the single-agent concept [7], multi-agent is applied as the individual clerk to settle down the conference date under the communication among each agent [8]; Four agents, Trainer Agent, the Neural Classifier Mobile Agent, the Interface Agent, and the Librarian Agent, retrieve documents satisfying a query and dealing with a specific topic [9]; using a multi-agent framework to do text classification automatically [10];apply the ant-clustering algorithm for text document classification[11].

Document classification is a kind of classifier training about the training set, and it analyzes the document requiring the classification according to the training data (classification knowledge base) and determines the category of the document. Currently, the two kinds of mostly used classifiers are KNN and SVM, both of which adopt the Vector Space Model to calculate each document vector. After calculating the document vector, KNN finds out the most similar document vector in the training set and then determines which category it belongs to. Meanwhile, when taking the classifier training, SVM will form a hyper plane to classify the different kinds of document vectors. Therefore, after finding out the document vector, the category of the vector is just the category of the document. And this kind of classification is in the statistical method, which calculates the weight of each term in each category of the document and then measures the weight of the term according to the different conditions of the words appearing in various categories. Hence, the category which can calculate the maximum weight of the term in the document is the category of the document. The concrete classification method will be detailed in the following chapters.

2.2 Term Weighting

In document classification, what we want to do is to classify documents according to the content of the document. In this thesis, we discuss text documents only. It's difficult to define the value of the document through the whole content. The easier way is to know which terms compose the content and to define the value of the document by these terms. Most researches define term weights by occurrence frequencies of terms or other term-weighting methods. Then we could generate the value, numbers/coordinates/vectors, of the document by term weights.

One of the term-weighting methods is TFIDF (Term Frequency * Inverse Document Frequency) [12]. TFIDF calculates the weight of the term and it combines term frequency (TF), which measures the number of times the term occurs in the document, and inverse document frequency (IDF), which measures the number of documents the term occurs. The inverse document frequency (IDF) is calculated as:

$$idf^t = \log \frac{N}{df^t}$$

The idf^t denotes the inverse document frequency of term t. idf^t denotes the number of documents the term t occurs and N is the total number of documents. TFIDF is defined as:

$$tfidf^t = tf^t * idf^t$$

where $tfidf^t$ denotes the weight of the term t. is the number of times the term t occurs in this document. The main idea of TFIDF is that when a term appears in each document, this term's recognition ability for the document is diminished. It cannot be used to distinguish one document from another.

Recently, an additional process is added. To analyze the meaning of the document and make similar documents have the close vectors or coordinates. The natural language processing, called Latent Semantic Analysis (LSA), is proposed in [13]. In information retrieval, it is sometimes called latent semantic indexing (LSI). One technique in the domain of LSI is Singular Value Decomposition (SVD) which is to do some matrix computation. The details of SVD is to transfer words of the document to the column of the matrix, called document matrix, and then compute the document matrix in the SVD way to generate three matrixes, U, S, and V. Then, matrix V multiplying matrix S could generate coordinates of all documents. By the property of SVD, similar documents have the close coordinates.

3 Intelligent Agent System Techniques

The Intelligent Agent System Mechanism is expected to imitate the human classification logically and thinking norm so as to deal with the large sum of data effectively. This kind of research focuses on “Intelligent Chinese Semantic Processing” and “Rule Inference Engine”, which can classify the materials according to the semantics of the materials or make its own classification rules according to the different requirements.

Autonomy: the agent can make use of the existed information to handle with the data and reach the working aim automatically without artificial operation. The document agent in this system is able to deliver the document to the right place through the classification mechanism.

Reactivity: the agent can make use of the interface of other users to produce some effects. When there are some wrong or correct information, the agent will collect the information and correct the wrong information automatically and improve the internal ability without influencing the surrounding operating systems. Therefore, this system can establish an Error Process Agent which can enhance the accuracy rate effectively through the mechanism.

Learning: the agent can automatically store the valuable data information as the reference for the new data so as to improve the accuracy rate and efficiency.

Collaborative: the agent can finish some complicated tasks together with the surrounding operating system. The intelligent document agent in the research will be described as follow:

3.1 Introduction of Artificial Document Handling Flow

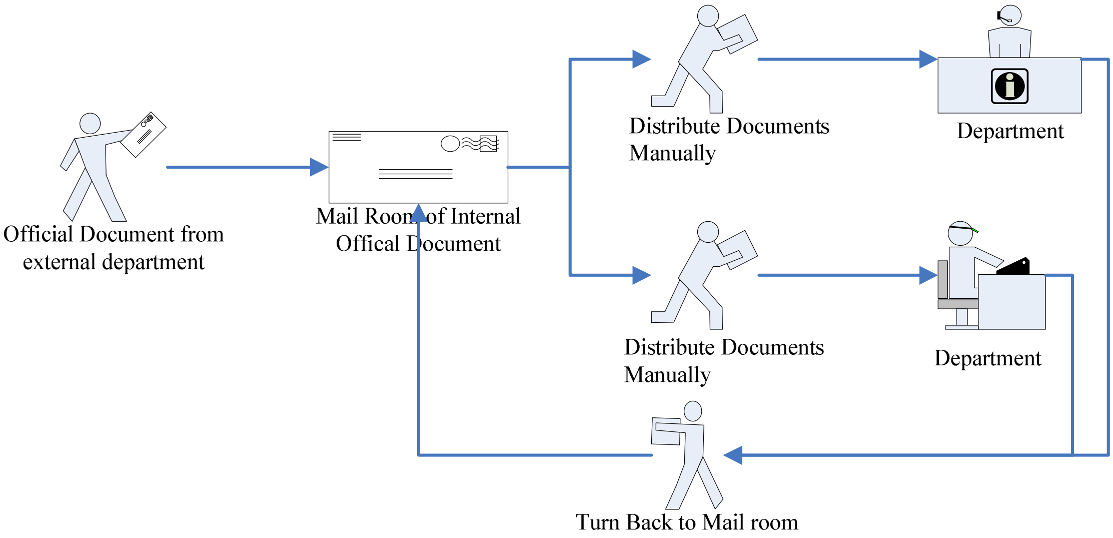

Official Document from external department
Mail Room of Internal Offical Document
Distribute Documents Manually
Department
Distribute Documents Manually
Department
Turn Back to Mail room

Fig. 1. Manual Document Handling Flow

In Fig. 1, the working flow without computer is that, when the document of external agency is sent to the receiving and delivering center, the document will be classified by the worker's experience and then be delivered to the agency manually. Usually, the document should be returned to the center when there is something wrong with the individual judgment, which may waste a lot of time; and then it relies on the individual judgment once again, and if there are some changes in staff composition, there may be some influences on the efficiency and accuracy of the document transmission.

3.2 Introduction of Intelligent Document Agent Flow

The following is the introduction of the relative characters of the Intelligent Document Agent System Flow Figure:

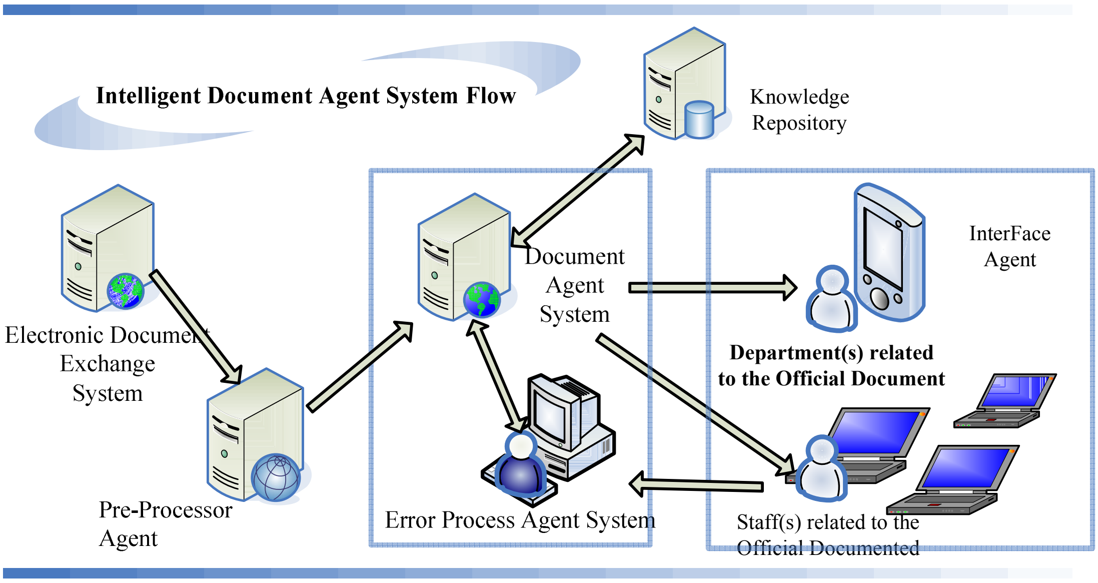

Intelligent Document Agent System Flow
Knowledge Repository
Electronic Document Exchange System
Document Agent System
InterFace Agent
Department(s) related to the Official Document
Pre-Processor Agent
Error Process Agent System
Staff(s) related to the Official Documented

Fig. 2. Intelligent Document Agent System Flow

Pre-Processor Agent: which stores the XML file and extract the attribute data required by XML, and send the longer data to the Chinese Sentence Segment System to cut it into shorter parts.

Knowledge: This can store the data which is necessary to the document agent and Error Process Agent, such as training set and testing set…

Document Agent System: The system for classifying documents.

Error Process Agent System: where the wrong classified documents or the ones that are unable to be classified are sent and are corrected artificially.

Interface Agent: which establishes some interfaces used after the classification, and which can provide other management like paperwork and processing, etc.

In Fig. 2, the system receives the electronic document data from the Electronic Document Exchange System and changes it into the required data format through lead processing, and then the data is classified by the Document Agent System. If it is unable to classify or it is wrongly classified, the data will be sent back to the Error Process Agent, and then the corrected information will be sent to the Document Processing Agent System. During this process, the wrong information of the Document Agent Classification must be corrected by the Classification Error Agent System and then the corrected information will be sent to the document agent and trained with the original correct information so as to improve the accuracy of the document agent.

3.3 Design of System Architecture

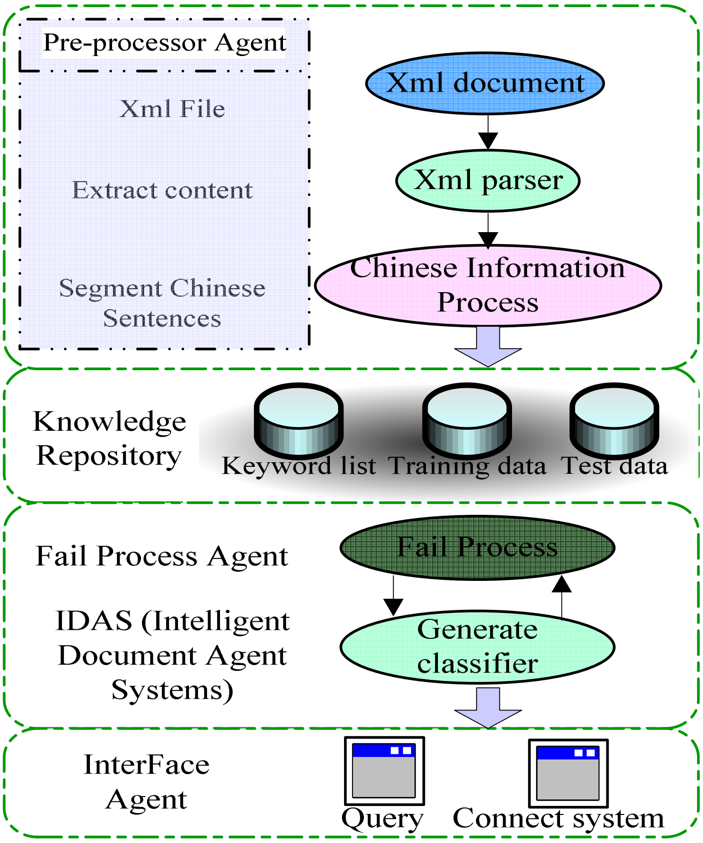


Fig. 3. System architecture

Preprocessor Agent: This extracts the valuable sentence information in the XML file sent from the external systems by parsing techniques. And then the parsed information will be cut to several parts to find the relative keywords by the Chinese Sentence Segment System.

Knowledge Repository: This stores the relative data received by Preprocessor Agent.

Fail Process Agent: when IDAS can't make automatic classification, the relative documents will be sent to this system. After processed by the individual experience, the data will be recorded and trained again for the aim to improve the accuracy.

IDAS (Intelligent Document Agent Systems): the main technique is to recognize the classification.

Interface Agent: Query Agent: which provides an User Inquiry System which can randomly check the rightness of the IDAS category by the individual experience and mark the wrong data and train them so as to improve the accuracy.

3.3.1 XML parsing

The use of library: Xerces

Xerces is a kind of parser set certificated by XML[1] and promoted by Apache organization, which provides a set of Application Programming Interface which is available to the program designer. To enhance the speed of program development, it combines C++ with JAVA to form an application programming interface for two parsers of W3C, which are DOM and SAX.

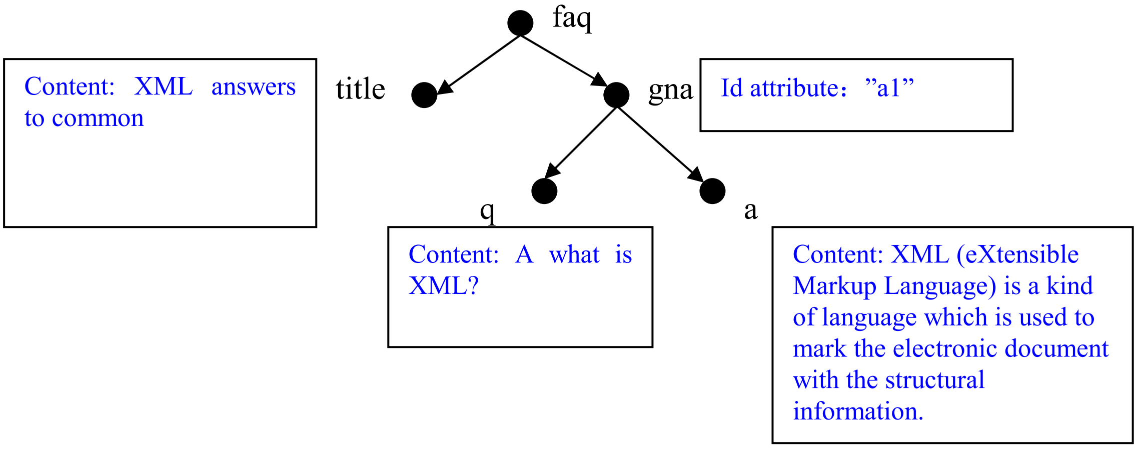
XML (eXtensible Markup Language) is a kind of language which is used to mark the electronic document with the structural information.

The following is an example for XML file:

```
  <?xml version="1.0" encoding="Big5"?>
<faq>
 <title>XML Answers to common questions </title>

 <qna id="a1">
  <q>A. What is  XML ?</q>
  What is XML?
  <a> XML (eXtensible Markup Language) is a kind of language which is used to mark the electronic document
with the structural information. </a>

 </qna>
</faq>
```


faq
title
gna
q
a
Content: XML answers to common
Id attribute："a1"
Content: A what is XML?
Content: XML (eXtensible Markup Language) is a kind of language which is used to mark the electronic document with the structural information.

In the tree data structure produced in the XML document, the dark point is the node, the word nearby is the name of the element node and the term in the rectangular frame is the data of the term node.

Under the understanding upon some concepts of XML and parser, we will continue to introduce Xerces. It supports SAX version 2 and DOM level 2, as well as SAX version 1 and DOM level 1.

3.3.2 Error Process Agent

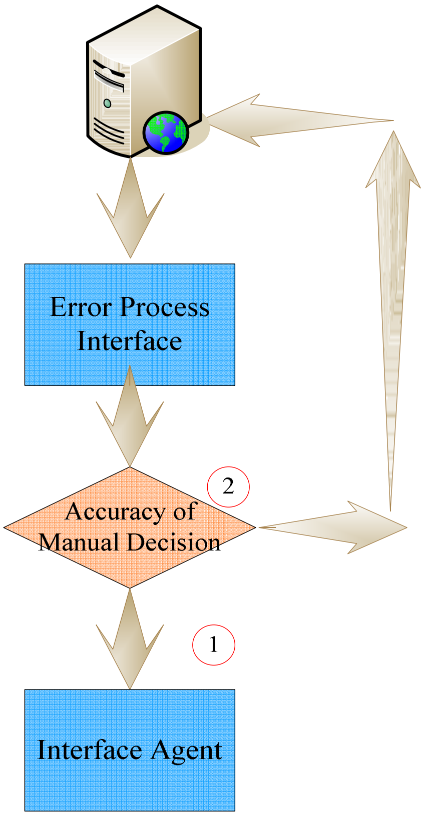


Fig. 4. Error Process Flow

When document agent sends the wrongly classified information or the one that is unable to classify to the Error Process Interface, the data will be judged artificially and delivered to the interface agent shown in 1st part of Fig. 4, and then the same materials will be sent to the document agent for some training as shown in 2nd part of Fig. 4. This can improve the classification accuracy of the document agent.

3.3.3 Design of IDAS (Intelligent Document Agent Systems) System Architecture

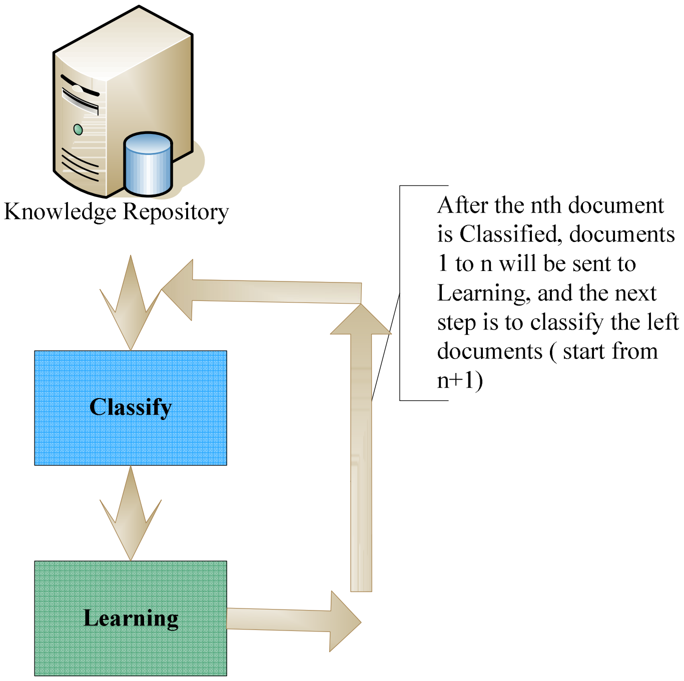

Knowledge Repository
After the nth document is Classified, documents 1 to n will be sent to Learning, and the next step is to classify the left documents (start from n+1)
Classify
Learning

Fig. 5. IDAS System architecture

- **Knowledge Repository**

XML Data (all the data in the broken documents) the new document to be classified, and its format is XML.

- **Classifier**

Calculating and summing up weights of the terms, generated at the previous step, in each category by our formula. The category which could calculate the max total weight is what the new document belongs.

- **Learning**

To increase one to the weights of terms in the document in the category which the system determines precisely; to decrease one to the weights of terms in the document in the wrong category which the system determines wrongly.

3.3.3.1 IDAS System Process Flow

Before the classification, the data should be divided into two parts, training set and testing set.

Step 1--- Knowledge Repository: First, it processes the data of XML file to produce the data set.

This research classifies the document according to its thesis and at first extracts the thesis data from the electronic document in XML format.

The extracted thesis data is sent to the Chinese Character Segment System developed by the central academia information science institution to be cut into several parts and then to obtain the broken Chinese terms.

It will calculate the weight of these Chinese terms with the term-weighting calculation method introduced in the following chapters.

It will produce the input format for the parser, which is the broken Chinese term.

Step 2---Classify: The data of Knowledge Repository is classified in the classification calculation method, which can calculate the score of the term in each category of the broken document, as well as the highest score of the category which the document belongs to. And it will get the correct category of the document according to the record in the database and calculate the accuracy of each classifier, compared with the classification results.

Step 3---Learn: Judging whether the classification is correct or not is to compare the classification results with the record in the database. This document has been classified, so it can be put into the training set, that is, the Knowledge Repository. When the third step is done, you can get another document from the testing set and follow the first step, putting the classified document into the Knowledge Repository of the classifier. The learning step in this research is to plus one point for the weight of the term in the category of document. But if the classification result is incorrect after the comparison, the weight of the thesis term in this wrong category will lose one point.

3.3.3.2 Classify Scenario of the Proposed System

Our method is based on the occurrence frequencies of the terms and the weighting rule to create the weight of each term in each category (Fig. 6). The weighting rule is our learning method. We will describe that later. The weight of the term in the category could be viewed as the importance of the term in the category. After retrieving terms from the new document, we sum up their weights in each category by our formula. The category the new document belongs to is what could calculate the maximum total weight. The reason is that the total weight calculated and summed up in each category is the importance of the new document in that category. For a document, the larger value of its importance in the category is, the more frequently the terms in it occurs in this category. And the terms which occur frequently in the document have positive effect to this category. We could infer that the document which containing these high weight terms belongs to this category which these high weight terms exist in. So we take it as our classification method that the category the document belongs to is what could calculate the maximum total weight.

After classifying, if the system could get the correct category the document belongs to, defined by humans, it will start the learning procedure. We separate our method into two steps, Classification and Learning. We describe them in the following individually.

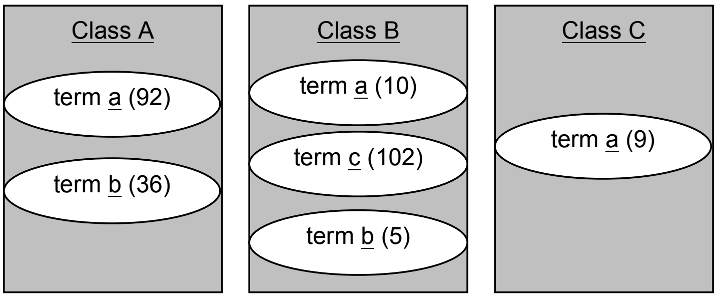


Fig. 6. Distribution of the classes with terms. "Term a(92) in Class A" means the score of the term a in the Class A is 92

3.3.3.2 Term-Weighting Calculation Method

Before setting up the classifier, we calculated the terms/words relation to classes, then removed the terms that wouldn't affect the accuracy. This paper is about classification according to three properties of official documents: subject, sender and document ID. After the introduction of the subject into a segmented Chinese system, the data/terms for classification can be isolated.

The data of the classification criteria were divided into sets, training set and testing set. The training set was analyzed by calculating the influence of terms/words then removing any terms/words that did not have any effect. This will reduce the possibility of error in the training set. The calculations of these influences are shown below:

$I_t = C/C_t$

I_t : The effect of t term

C : Number of classification

C_t : The number of classification that contains t term

This idea is similar to IDF (Inverse Document Frequency). The only difference is in the denominator of the formula.

In IDF, the denominator is the number of documents that contain certain terms. In our formula, C_t occurrences increase, meaning term t is not a key to a single, certain classification but the possibility that term t is a common word.

Such is the basic principle of our analysis.

3.3.3.4 Classification Module

Our term-weighting strategy is based on occurrence frequencies of terms in the category because the occurrence frequency of a term is relative to its importance in that category. And our classifying strategy is to calculate the square root of each term weight in order to narrow down the difference among weights of terms. Then, the square root of the term is divided by the number of documents this term occurs. This division makes the importance of this term lowered because the more categories it occurs in, the less significant it is. Considering the concept of longest matching, additional weights are given to continuous terms. Total weights of terms in this new document are summed up in each category, and the category with the highest weights is identified as what this new document belongs. We introduce our classification formula first, and then explain the details of each process.

Definition 1: Classification Formula

$$\sum_{j=1}^{T}\left(\left(\sum_{i=1}^{Lj}\frac{\sqrt{S_c^i}}{C_i}\right)e^{(Li-1)*0.1}\right)$$

T : There are T continuous terms.

L_j : Number of terms consisting of the j-th continuous term.

S_c^i : Weight of the i-th term in the continuous term in the c class.

C_i : Number of classes the i-th term in the continuous term exists.

Definition of the continuous term: in this class, if the term b is next to the term a, and they exist in this class, then we can say that the term a and b consist of the continuous term.

Analyzing documents, we found that the high frequent terms and low frequent ones are both significant. Although the contents of documents almost are different, some of them have similar purposes and have several common terms. So documents with similar purposes may produce high frequent terms, and documents which appear rarely will only generate few low frequent terms. High frequent terms are not absolutely significant, and low frequent ones are not absolutely valueless. We also found that the text length of the purpose in the document is shorter than the text of the content in the general document. If we use the common term-weighting scheme, the result won't be good enough. Take TFIDF as an example. TF is the frequencies of the term and IDF is log (all documents / number of documents the term occurs in). The text in the purpose tag in the document is short and brief, and the terms in the text are almost different, occurring once in a document. So the value of TF is 1, which could be ignored. But considering the value of IDF, the term which occurs frequently lowers its value of IDF. That means it would lower the importance of the term which occur frequently. That is not always true in our environment.

Our term-weighting strategy is to calculate the square root of the term weight in order to avoid what TFIDF causes. The square root of the weight narrows down the difference among weights of the terms in every category. When calculating total weight in each category, the operation of square root makes weights of the terms which distribute averagely in the category larger than weights of the terms which differ too much in the category. Unless the difference in the category where the terms differ much is big, or the average value of the weights is small in the category where weights of the terms distribute averagely. However, it tells us that the category with most weights of the terms above average has a chance to beat the category with a few higher/highest weights of the terms. Weights of the terms which distribute averagely in the category are higher means these terms are more significant. And high weights of the terms are a few in the category where weights of the terms differ much means only a few terms are significant. We focus on these two situations:

--The high weights of the terms should be high enough in the category where weights of the terms differ much.

--The average of weights of the terms should be high in the category where weights of the terms distribute averagely.

These two the situations make the total weight summed by our formula in the category heaviest. And they are our favorite categories determined and also closer to the reality.

Other operations in our formula are to divide C_i and to add the bonus for the continuous terms (to multiply $e^{(Lj-1)*01}$). The more categories the term occurs in, the less significant the term is to the categories. So we should lower the weight of the term, that's to divide C_i. Then we explain the significance of continuous terms. If we receive an unknown sentence, the more terms we know in that, the more we understand its meaning. And if the terms, we know, are continuous and consist of a part of the sentence, we could understand the unknown sentence more (Fig. 7). Based on this rule, if a category recognizes continuous terms in the content in the document, we would give the bonus, multiplying $e^{(Lj-1)*01}$, to these continuous terms in this category in order to increase their weights.

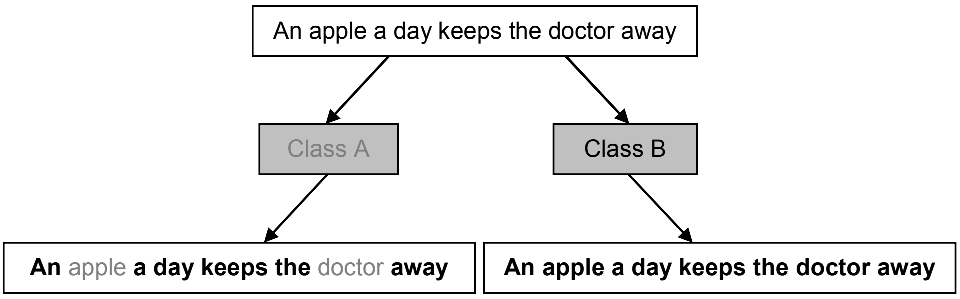

An apple a day keeps the doctor away
Class A
Class B
An apple a day keeps the doctor away
An apple a day keeps the doctor away

Fig. 7. Terms in Class B consist of the longer sentence. It is more possible for Class B to understand "An apple a day keeps the doctor away"

3.3.3.5 Learning Module

Our learning strategy is as follows: if our method classifies the document correctly as the right category, we increase one to the weights of the terms in this document in this right category; if our method classifies as the wrong category, we decrease one to the weights of the terms in the document in this wrong category, and we increase one to the weights of the terms in the document in the category the document belongs to (Fig. 8). That is the knowledge learning of our classifier. Our method classifies the document wrongly, that means the total weight (in the wrong category) is larger than one in the category the document belongs to. To compensate the effect of classification errors that terms are classified into the wrong category, we make the corresponding term weights decreased. If the total weight summed in the correct category is largest, that means these terms are able to stand for this category and could be identified as this category. In order to keep the identification of these terms, we increase their weights as a reward.

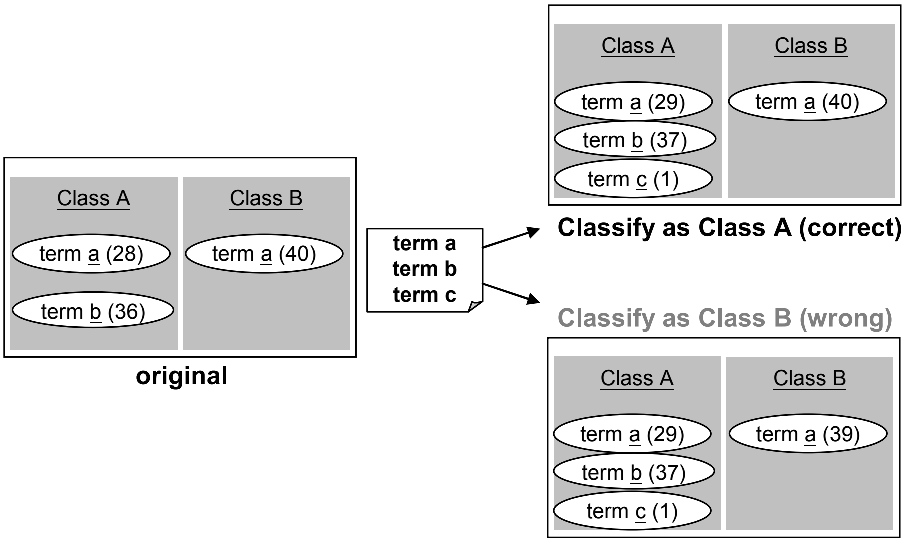


Fig. 8. Learning strategy to revise term weights

3.3.3.6 System Module and Introduction

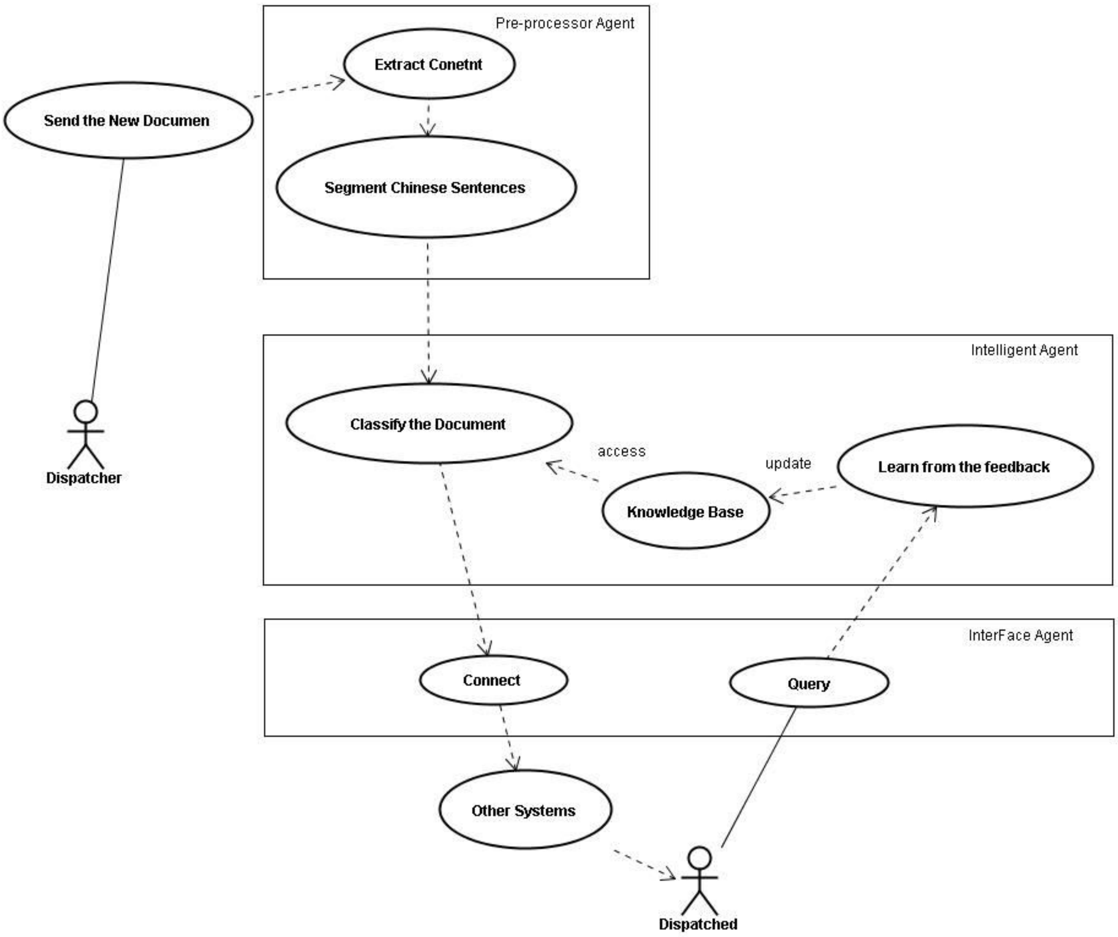


Fig. 9. System Module Flow

The dispatcher first sends the new document to Pre-processor Agent. The Extract Content of the Pre-processor Agent will extract the suitable contents and then the Chinese Sentence Segment System will segment the extracted terms; when all is done, it will send the document to Intelligent Agent. The Intelligent Agent will classify the document according to the relative weight of the received contents and classify them with the method mentioned above. And you will get the category of the document and then the result will be sent to the Interface Agent. To integrate with other systems, the content in the Interface Agent will send the processing result to the relative system in some follow-up processes, that is, to send the document to the relative dispatcher.

If considering the result is incorrect, the dispatcher will send the correct result back to the Intelligent Agent through the Query of the Interface Agent. And to learn from the Feedback means that the Intelligent Agent learns and updates the Knowledge Base according to the feedback from the dispatcher.

3.3.4 InterFace Agent

Interface, means the communication between people and machine, including the display and input device in its narrowing concept. In the Software Ergonomics of Information Science, the dispatcher will learn the program interacted with people through the interface. The Interface Agent refers to whether the computer can provide services actively, but not the format of the interface. The current interface between people and machine is usually done under the direct manipulation; on the contrary, if acting with the Interface Agent, the computer will think for the dispatcher and provide the active services.

The main work of the Dispatcher Interface Agent is to interact with the dispatcher and promote other agents. Therefore, when the document has been sent to the dispatcher, the Interface Agent can provide the dispatcher with the process and inquiry materials. On the inquiry interface, you will get the clear information about the document quantity needed to be handled with and on the process interface, you will be able to process the document, as well as establish a working flow system to get rid of the relative checking and signing activities. Meanwhile, the system can help to save a lot of artificial processing, so it can not only reduce the personnel burden but also improve and enhance the document-processing efficiency.

4 Discussions and Future Works

It wastes both time and energy to distribute the documents artificially. The staff has to distinguish the units the documents belong to by the personal experience, and it is easy to make a certain mistakes during this kind of transmission; if there is anything wrong, the documents will be returned and distributed again. Hence, this method costs a lot of time and personnel, and it is also not easy to make the tracking inquiries. However this system can deal with this problem very well, what's more, the artificial transmission lasts one to two days, and this system only needs less than 20 seconds to finish the transmission, the efficiency of which has been greatly improved.

Right now, the testing range of this research chooses such a kind of system as the testing example, and longs to enlarge it to other systems with the similar functions in the future, such as the automatic classification of the online test papers and how to improve the system's application range.

As to the Intelligent Document Agent, it is expected to improve its calculation and find out a better means to enhance the efficiency and accuracy, so that it is completely able to get away from the artificial processing.

It is expected that in the future it can reach how to develop the working flow system and let the classified data pass the working distribution. Meanwhile the system is capable of controlling the processing procedure so that it can substitute the artificial method.

5 Conclusions

A great challenge in the research on a theoretical and practical agent systems, is how to find the classification method of the highest efficiency and accuracy, and set up an application system based on the high techniques. In fact, this could improve the working flow efficiency and produce some results similar to the individual experience method. This research provides a set of document agent which can be applied in artificial distribution of documents; and this system can also link the theory and fact together by using some relative techniques of document classification.

As to the Intelligent Document Agent, it adopts the statistic classification method. The classification process is very short and can be learned through the machine, so it can improve the accuracy of the classifier. The average accuracy rate of the classification results can reach 80%. Although this result still leaves some spaces to develop, it represents the ability to reduce 80% of the personnel burden and improve the convenience of the working flow greatly.

References

[1] XML, *http://www.w3.org/XML/*

[2] A. Hector, V. L. Narasimhan, "A New Classification Scheme for Software Agents", in *Proceedings of the 3rd International Conference on Information Technology and Applications*, pp.191-196, 2005.

[3] Y. Sun, H. Peng, Z. Lin, D. Hu, "A Software Agent System for News Information Delivery on Internet", in *Proceedings of the 23th International Conference on Information Technology Interfaces*, pp.151-156, 2001.

[4] E. M. Duarte, A. P. Braga, J. L. Braga, "Internet economic news gathering and classification: a neural network software agent based approach", in *Proceedings of VII Brazilian Symposium on Neural Networks*, p.112, 2002.

[5] I. Khan, H. C. Card, "Personal Adaptive Web agent: a tool for information filtering", in *Proceedings of IEEE 1997 Canadian Conference on Electrical and Computer Engineering*, pp.305-308, 1997.

[6] J. Chen, Y. Wu, P. C. Y. Sheu, M. Li, B. Hui, "Bayesian Classification-based Intelligent-agent Data Management over Grid", in *Proceedings of 2006 IEEE International Conference on Tools with Artificial Intelligence*, pp.93-97, 2006.

[7] S. Peng, S. Mukhopadhyay, R. Raje, M. Palakal, J. Mostafa, "A Comparison Between Single-agent and Multi-agent Classification of Documents", in *Proceedings of Parallel and Distributed Processing Symposium*, pp.935-944, 2001.

[8] P. J. Modi, P. W. T. Kim, "Classification of Examples by Multiple Agents with Private Features", in *Proceedings of 2005 IEEE International Conference on Intelligent Agent Technology*, pp.223-229, 2005.

[9] G. Pilato, S. Vitabile, "A Neural Multi-Agent Based System for Smart Html Pages Retrieval", in *Proceedings of 2003 IEEE International Conference on Intelligent Agent Technology*, pp.233-239, 2003.

[10] J. Mostafa, W. M. Ke, Y. Y. Fu, "Automated Text Classification Using a Multi-Agent Framework", in *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pp.157-158, 2005.

[11] A. L. Vizine, L. N. de Castro, R. R. Gudwin, "Text Document Classification Using Swarm Intelligence", in *Proceedings of 2005 IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pp.134-139, 2005.

[12] G. Salton, M. J. McGill, *Introduction to modern information retrieval*. Mc-Graw Hill, New York, NY, USA, 1983.

[13] S. Deerwester , S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, "Indexing by Latent Semantic Analysis", *Journal of the American Society of Information Science*, Vo.41, No.6, pp. 391-407, 1990.