Available online at www.sciencedirect.com
ScienceDirect
Computers & Education 51 (2008) 337–353
COMPUTERS & EDUCATION
www.elsevier.com/locate/compedu

An enhanced genetic approach to optimizing auto-reply accuracy of an e-learning system

Gwo-Jen Hwang [a,*], Peng-Yeng Yin [b], Tzu-Ting Wang [b], Judy C.R. Tseng [c], Gwo-Haur Hwang [d]

[a] *Department of Information and Learning Technology, National University of Tainan 33, Sec. 2, Shulin Street, Tainan City 70005, Taiwan, ROC*
[b] *Department of Information Management, National Chi Nan University, Pu-Li, Nan-Tou 545, Taiwan, ROC*
[c] *Department of Computer Science and Information Engineering, Chung-Hua University, Hsinchu 300, Taiwan, ROC*
[d] *Information Management Department, Ling Tung University, Taichung 40852, Taiwan, ROC*

Received 8 January 2007; received in revised form 10 May 2007; accepted 25 May 2007

Abstract

With the rapid development in Information Technology (IT), the Internet has become one of the central media for conducting learning. However, most of the existing web-based learning systems only provide stand-alone subject materials for browsing and may face some drawbacks. For example, if students encounter problems during the learning process, their learning performances could be significantly devastated due to no instant aid. As an on-line learning system may interact with thousands of students, it is almost impossible for the teachers or the teaching assistants to answer all the students' questions manually, which is not only inefficient, but also human laborious. To cope with this problem, an e-learning system that is able to automatically answer the students' questions on the fly based on the training cases given by the teacher will be presented in this paper. Moreover, an enhanced genetic approach is proposed to optimize the weights of keywords for each candidate answer according to the feedbacks provided by the students, hence more accurate answers can be provided in the future. Experimental results have shown that the developed system can provide more accurate answerers than existing approaches by employing the self-adjusting method.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: Computer-assisted learning; Web-based learning; Call center; Genetic algorithm

1. Introduction

Recent progress of computer and network technologies has encouraged the development of web-based learning environments (Hwang, 2003, Hwang, Lin, & Lin, 2006; Tseng & Hwang G.J., 2004), in which students can proceed their learning activities without being limited by location and time. However, as most of

* Corresponding author. Tel.: +886 915396558; fax: +886 63017001.
E-mail address: gjhwang@mail.nutn.edu.tw (G.-J. Hwang).

0360-1315/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2007.05.014

the existing web-based learning systems simply provide subject materials for browsing, the students are likely to be stuck while encountering problems during the learning process without instant aid, and hence their learning performances could be significantly deteriorated (Jonassen, 2000).

Some researchers attempted to employ on-line discussion groups to cope with the problem. Nevertheless, most of the answers obtained from the discussion groups could be incorrect or incomplete; therefore, the most desirable approach is to obtain the answers from the teacher. Unfortunately, for a popular on-line course with thousands of students, it is almost impossible for the teacher to answer every question submitted by the students, not to mention the provision of instant aids to them. Rau, Chen, and Chin (2004) indicated that, without face-to-face interaction, it is important to provide immediate help and interactions while proceeding on-line instruction.

To cope with this problem, a self-adjusting virtual tutoring assistant system is proposed in this paper. This system is able to answer students' questions automatically based on the training cases given by the teacher. Moreover, the system is able to accept the students' feedbacks and adjust the weights of keywords for each candidate answer, and hence more accurate answers can be provided in the future. Experimental results have shown that, the system can provide more accurate answerers by employing the self-adjusting approach. Moreover, as such systems offer student question–answering service 24 h a day, learning performance of the students can be improved by minimizing their frustration during the learning process in time.

2. Background and literature review

On-line learning and testing systems have become one of the most widely employed media for achieving various educational purposes due to the rapid progress in Information Technology (IT). In an on-line learning environment, students may participate the learning in different times and locations; that is, they may encounter a problem during the learning without instant help from the teachers. The learning process might get stuck and discourage the students from continuingly using the on-line learning systems. Therefore, it becomes an import issue to develop student service systems that can assist the student to solve the problem by automatically locating the feasible answers (Tseng & Hwang, 2006). Such student service systems are similar to the call centers or customer service systems in the business application domain.

Since 1985, enterprises started to build up call centers to provide customer services. Most of the call centers only serve consumers at regular office hours. Moreover, the service personals must receive a series of training courses before they can offer appropriate services to the customers. This implies that a requirement of a large amount of training cost will be incurred by the establishment of the call centers. It can be seen that such traditional service systems not only provide inefficient and ineffective service, but also increase the service cost; therefore, the development of automatic customer service systems has become an important issue for enterprises. Witt, Andrews, and Carlson (2004) examined the relationship of the interaction between emotional exhaustion and conscientiousness with objectively-measured call volume performance and subjectively-measured service quality ratings among 92 call center customer service representatives of a financial service institution. They indicated that the interactive effects on call volume but not service quality; therefore, it might be a good idea to develop automatic customer service systems.

A call center and its associated Information Technology (IT) provide an opportunity to redesign and improve service-delivery operations (Adriaa & Chowdhury, 2004). On-line FAQ databases are frequently adopted in traditional customer service systems, especially in the Internet environment (Burnetta & Bonnici, 2003). Many technical companies not only employ on-line call centers to improve the service quality, but also provide relevant techniques and software to assist other companies in maintaining customer relationships; for examples, the call center systems of Ticali, Dell and IBM.

Although providing FAQ database can reduce service cost, the web site users need to select the possible category, link to the database and search for the answers manually, which is usually time consuming, and the customers are likely to reach the limits of their patience, especially when the system load is heavy or the network traffic is burdened with a large amount of requests.

To more accurately identify the requests of customers, researchers have proposed several methods in recent years. For example, Hoch (1994) presented the statistical methods of information retrieval used at INFOCLAS, which is capable of classifying print business letters according to message types such as order, offer and enclosure. In 1996, Cohen (1996) proposed the "key word spotting rule" approach, which can efficiently classify e-mails and has been applied to the development of e-mail management systems. In the meantime, Cooper (1996) reported the FAQfinder, which employed a set of weighted parameters to determine the similarity between the customer's request and the FAQ. Later, Li and Tseng (2001) proposed an Intelligent Network-based Customer Service System (INCSS), which assigns a weighted keyword set to each FAQ, and then compares the keyword set of the customer's request and that of each FAQ to find the most feasible answer. INCSS can automatically reply the requests submitted from the customers.

Recently, Tseng and Hwang, 2006 proposed an Automatic Customer Service System (ACSS), which classifies FAQ documents based on six interrogatives, i.e., "who", "what", "where", "when", "why" and "how", to improve the performance for finding the most feasible answers. While receiving a request, ACSS will generate a CV for the request, and compares it with those of the FAQ's by employing the space vector model to find the most feasible answer for the customer. If no feasible answers can be found, ACSS will forward the inquiry to the appropriate service staff. Once the service personnel has provided an answer to the request, the new request as well as its answer will be recorded in the FAQ database, such that the system will be able to automatically reply similar requests in the future.

To develop an effective auto-reply system, the issue of information retrieval has attracted the attentions from researchers in this field. Spink (1995) studied the retrieval effectiveness of search terms identified by users and intermediaries from retrieved items during term relevance feedback. He found that terms from the users' question statements were the most effective among others. Later, Savoy (1996) proposed vector-processing scheme for searching information in hypertext systems. Vakkari (1999) analyzed certain features of work tasks and relate these features to types of information people are looking for and using in their tasks, patterning of search strategies for obtaining information and relevance assessments in choosing retrieved documents. Recently, Kerenidis and de Wolf (2004) presented an algorithm to cope with the Private Information Retrieval problem. Moreover, Hansen and Jarvelin (2005) presented empirical results to show that the patent task performance process involves highly collaborative aspects throughout the stages of the information seeking and retrieval process. They also showed that these activities may be categorized and related to different stages in an information seeking and retrieval process.

Other relevant studies include the issues of keyword retrieval and sentence similarity comparison. There are several ways to retrieve keywords, e.g., term extraction method (Salton & Buckley, 1998), phrase extraction method (Krulwich, 1995), and Statistic analysis method (Jones, Gassie, & Radhakrishnan, 1990). Term extraction method can detect important terms from classified input text; phrase extraction method is used to detect phrases in the text; statistic analysis method can identify possible keywords from a large amount of unclassified input text by computing the occurrences of each keyword. Among those proposed methods, $TF \times IDF$ is a well-known scheme for ranking N documents according to their relevance to a query containing M query terms (Salton & McGill, 1983). Researchers have attempted to extend the scheme to extract additional information from hypertext links to enhance retrieval effectiveness in the World Wide Web environment (Savoy, 1996; Yuwono & Lee, 1996). Let the N documents be D_i, $1 \leqslant i \leqslant N$, and the M query terms of a query Q be Q_j, $1 \leqslant j \leqslant M$. The original $TF \times IDF$ formula for determining the relevance between D_i and Q is given as follow.

$$R(D_i, Q) = \sum_{Q_j \in Q} \left(0.5 + 0.5 \frac{TF(D_i, Q_j)}{TF_{\max}(D_i)}\right) IDF(Q_j) \quad \text{and} \quad IDF(Q_j) = \log\left(N \Big/ \sum_{i=1}^{N} C(D_i, Q_j)\right) \tag{1}$$

where $TF(D_i, Q_j)$ represents the number of occurrences for Q_j in D_i, $TF_{\max}(D_i)$ indicates the maximum number of occurrences for the key terms in D_i, and $C(D_i, Q_j) = 1$ if D_i contains Q_j; $C(D_i, Q_j) = 0$, otherwise.

3. Definition of auto-reply accuracy optimization problem

Among various issues concerning information retrieval, a central problem is to rank the archived documents according to their relevance to a submitted query (Larkey & Connell, 2005). Croft (1987) suggested that significant improvements in retrieval performance will require techniques that, in some sense, "understand" the content of documents and queries in order to infer their probable relationships. According to this view, information retrieval is an evidential reasoning process in which the probability that a user's information need is met given a document as "evidence" can be estimated (Syu & Lang, 2000).

The following auto-reply scenarios are considered in this paper. Assume that a student submits a question Q comprised of $m1$ keywords, $K_i(Q)$, $i = 1, \ldots, m1$, to an auto-reply system which is connected to a repository of n document, $D_1, D_2, \ldots, D_n$. Let each document D_q be characterized by a document vector containing $m2$ keywords, $K_j(D_q)$, $j = 1, \ldots, m2$. Also let $w(K_j(D_q))$ be the relevance weight of the jth keyword in D_q. The similarity between the question Q and the document D_q is computed as follows:

$$FN(K(Q), K(D_q)) = \sum_{i=1}^{m1} \sum_{j=1}^{m2} \left(\text{equals}(K_i(Q), K_j(D_q)) \times \frac{w(K_j(D_q))}{\sum_{r=1}^{n} w(K_r(D_r))} \right) \quad (2)$$

where equals$(K_i(Q), K_j(D_q)) = 1$ if keyword $K_i(Q)$ is equal to $K_j(D_q)$; 0, otherwise.

For question Q, the document D_q with the maximum value $FN(K(Q),K(D_q))$ will be retrieved as the best-fit answer to that question. To ensure the best-fit document to the questions submitted by the student can be retrieved, a set of training cases is collected to adjust the weights of the document keywords. Moreover, a set of test cases is collected to test the accuracy and the applicability of the matching mechanism based on the trained weights.

Each training case or test case consists of a question submitted by the student and a link to the best-fit document identified by the teacher. Note that for most FAQ databases, the questions and the corresponding best-fit links to the documents have been well recorded during the FAQ collecting process; therefore, only few manual efforts are needed in preparing the training set and the test set. Assume that there are n historical documents collected in the database and a one-to-one and onto mapping function $f{:}\{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots n\}$ where $f(p) = q$ indicates that the domain-experts (teachers) label document D_q as the best-fit answer to question Q_p. The goal of the Auto-Reply Accuracy Optimization (ARAO) Problem is as follows:

$$\text{Maximize} \sum_{p=1}^{n} (\underset{q}{\text{Max}}\, FN(k(Q_p), k(D_q)) \times \text{Match}(Q_p, D_q)) \quad (3)$$

where Match$(Q_p,D_q) = 1$ if $f(p) = q$, and Match$(Q_p,D_q) = 0$, otherwise.

The ARAO problem aims to find the optimal keyword weights $w(K_j(D_i))$ that give the best-fit answers to the maximum number of the questions in the training cases.

4. Genetic-based keyword weight optimization

In this section, a genetic-based auto-reply system is proposed. This system consists of several components, including keyword abstraction, weight initialization, keyword matching, and genetic weight optimization. For each course, a keyword database that contains a set of domain-relevant keywords is maintained by the teacher. Each document in the FAQ database is preprocessed by extracting the words that are identified in the keyword database and forming a document vector to characterize it. Once a question has been submitted by the learner, the auto-reply system will try to identify the features of the question by checking if any word of it is in the keyword database. The matched keywords of each question form a question vector, which represents the features of the question, and is then used to find the best-fit answer by applying the similarity comparison formula addressed in Eq. (2). Before presenting this system, some basic principles of genetic algorithms are briefly introduced.

4.1. Basics of genetic algorithm

Evolutionary computation (EC) is a category of algorithms which are developed according to Darwinian's rule "survival of the fittest" and take metaphor of natural evolution. EC serves as an optimizer to complex problems and has been applied successfully to many applications. Early ECs include evolutionary programming, evolution strategy, genetic algorithms, and genetic programming (Han & Kim, 2002; Toffolo & Lazzaretto, 2002). More recently, ant colony optimization, particle swarm optimization, and adaptive systems have also been proposed.

Genetic algorithm (GA) (Holland, 1975) is one of the most intensively-used techniques from EC. According to Darwinian's rule, fitter individuals will have a higher probability to survive and pass their genes to the next generation through genetic operations. GA simulates Darwinian's rule and has manifested successful applications such as scheduling (Murata, Ishibuchi, & Tanaka, 1996) and time versus cost optimization (Feng, Liu, & Burns, 1997).

The GA solves an optimization problem in a systematic way. The standard GA consists of several components, namely chromosome encoding, fitness evaluation, selection, crossover, and mutation. The flow diagram of a standard GA is shown in Fig. 1. Each component is described in the following:

- *Chromosome encoding and population initialization*
  Chromosome encoding indicates the process which transfers a candidate solution to the optimization problem into a chromosome which consists of several genes. In GA, a gene could be a binary bit, an integer, a real number, or even a symbol, depending on the underlying optimization problem.GA fosters a population of chromosomes. The initial population is usually generated at random instead of using a problem-specific heuristic in order to prevent the modeling bias, and the population evolves from generation to generation through a number of genetic operators such that the quality of the population improves. The population size is kept constant throughout the whole evolution.
- *Fitness evaluation*
  According to Darwinian's selection rule, the fitter individuals have a higher probability to be selected into the next generation. To evaluate the fitness of each chromosome, a *fitness function* should be defined to assess how well the solution represented by the chromosome solves the addressed problem. Usually the objective function of the underlying application can be used for this purpose. However, when the generated solutions hardly satisfy the constraints imposed by the application, a penalty function can be incorporated into the objective function to reflect the fitness of the solutions.
- *Selection*
  Selection is a process which mimics the natural survival of the fittest individuals. Each chromosome in the population is associated with a fitness by applying the solution represented by the chromosome to the fitness function. The probability with which a chromosome is to be selected is proportional to its fitness.

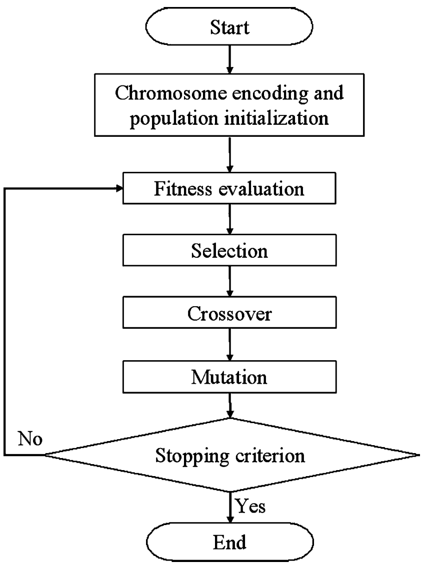

Fig. 1. Flow diagram of a basic GA.

Among others, roulette wheel selection is one of the popularly used selection schemes and proceeds as follows. Let f_i be the fitness of chromosome i, the selection probability S_i of chromosome i is defined as $S_i = f_i/\sum_{j=1}^{P} f_j$ where P is the population size. To perform the selection according to the probability distributions of S_i, the accumulated selection probability A_i is computed by $A_i = \sum_{j=1}^{i} S_j$. Then, a random real number $q \in [0.0, 1.0]$ is drawn from a uniform distribution. The roulette wheel selection will pick chromosome i if $A_{i-1} < q \leqslant A_i$. The selection process will be repeatedly performed until the next population is fully filled.

- *Crossover*
  Crossover is a genetic process in which each individual has a chance to interchange gene information with its mate randomly chosen from the same population. Since the highly fit chromosomes take a large proportion of the population due to the selection process, they will receive more trials of crossover and extend the exploitation in "good" solution regions sketched by these chromosomes. Crossover is performed with a crossover probability. A random number can be generated between [0.0, 1.0] for each mating pair, if the random number is less than the crossover probability, the crossover is performed to produce offspring; otherwise, the mating parents are retained, i.e., no crossover is performed. There are two broadly employed crossover operators (Syswerda, 1989). The single point crossover yields offspring by interchanging all the genes after a random position from the parent chromosomes (see Fig. 2a). On the other hand, the two-point crossover generates two random positions and interchanges the genes between the two positions from the parent chromosomes as shown in Fig. 2b.
- *Mutation*
  Selection and crossover can only explore the chromosome space of existing genes and will get trap in a local optimum since no new genes are introduced. On the other hand, mutation is an occasional alteration of existing genes and is performed with a very low mutation probability. Mutation guarantees a non-zero probability of the search to any feasible chromosomes. If a gene is determined to perform mutation, its gene value is replaced by any feasible content given randomly. For the binary chromosome encoding case, mutation can be simply performed by flipping the bit values at the mutated gene positions as shown in Fig. 3.
  Traditional auto-reply systems determine the keyword weights solely based on the frequency information that the keywords appear in various documents. However, the relevance between the keywords and docu-

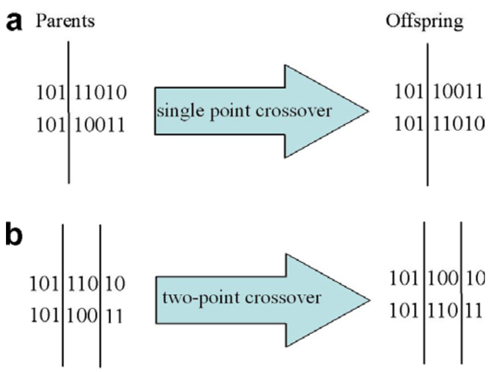

Fig. 2. An illustrative example of crossover. (a) Single point crossover, and (b) two-point crossover.

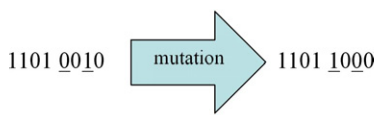

Fig. 3. An illustrative example of mutation where the mutated gene positions are underlined.

ments could depend not only on frequency but also on other factors such as semantics. In order to optimize the accuracy of an auto-reply system, a genetic approach to optimize the keyword weights is proposed in the following section.

4.2. Enhanced genetic approach to optimizing the weights of keywords

The enhanced genetic approach is based on the vector space model that is broadly used in the information retrieval. Both of user question and stored documents are represented as real-valued vectors in the real vector space. Each question/document vector is composed of a number of elements whose values are the weights of the corresponding keywords in the question/document. The similarity between a question and a document is estimated by a distance measure between the two vectors. Thus, the keyword weights play a central role in the accuracy obtained from the retrieved documents. Fig. 4 depicts the flow diagram of using the enhanced genetic approach for optimizing the keyword weights. The detailed steps are given in the following:

Input: The domain-experts provide n training question cases. Moreover, the domain-experts are asked to provide a one-to-one and onto mapping function f:$\{1, 2, \ldots, n\} \rightarrow \{1,2, \ldots, n\}$ between the training questions and the answer documents where $f(p) = q$ indicates that the domain experts label document D_q as the best-fit answer to question Q_p. Thus, a ground-truth question-to-document matching matrix Match(Q_p, D_q) is available where Match(Q_p, D_q) = 1 if $f(p) = q$, and Match(Q_p, D_q) = 0, otherwise. Output: Optimal document keyword weight matrix, $\mathrm{CVD}_q = [W_{q,1}, W_{q,2}, \ldots, W_{q,m2}]$, $q = 1, \ldots, n$.

Step 1. Segment the n training questions into keywords according to the keyword database and construct the keyword vector for each question, $\mathrm{CVQ}_p = [Q_{p,1}, Q_{p,2}, \ldots, Q_{p,m1}]$, $p = 1, \ldots, n$.
Step 2. The genetic approach randomly generates an initial population of chromosomes. Each chromosome is a candidate document keyword weight matrix containing $n \times m2$ real-valued elements. In order to expedite the convergence process, one of the chromosomes is encoded by taking the weight values determined using the classic $TF \times IDF$ model (see Eq. (1)). The other chromosomes take the weight values between [0.0, 1.0] generated at random.
Step 3. Evaluate the fitness of each chromosome using the fitness function (see Eq. (3)).
Step 4. Perform roulette wheel selection to generate the next population.
Step 5. Perform two-point crossover to generate the offspring.
Step 6. Perform mutation to introduce potential new genes into the population.

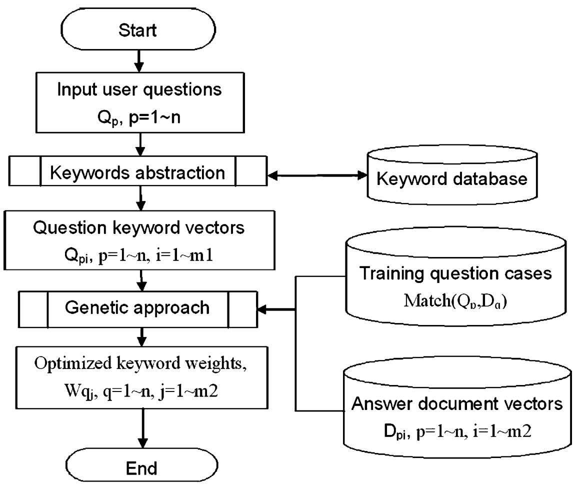

Fig. 4. The flow diagram of the genetic approach for optimizing the keyword weights.

Step 7. Repeat Steps 3–6 until experiencing a given number of generations or when the best fitness observed so far is larger than a predefined threshold.

4.3. An illustrative example

In this subsection, an illustrative example is given to demonstrate how the enhanced genetic approach works to adjust the weights of each document's keywords.

4.3.1. Optimizing document keyword weights

Taking the course of "Expert Systems" as an example, assume that the training cases contain five questions and five answer documents as follows:

$Q_1 =$ "What is constructive knowledge?"
$Q_2 =$ "What are the properties of procedural knowledge?"
$Q_3 =$ "What is strategic knowledge?"
$Q_4 =$ "What is semantic knowledge from the learner's point of view?"
$Q_5 =$ "What are the four categories of knowledge classified based on the learner's point of view?"
$D_1 =$ "The constructive knowledge indicates what kind of knowledge has been derived and accumulated via interactions among other people and this knowledge is classified based on the learner's point of view."
$D_2 =$ "Procedural knowledge indicates a set of step-by-step operations that follow a specified sequence to achieve the goal."
$D_3 =$ "Strategic knowledge is the knowledge derived and accumulated via a series of learning activities."
$D_4 =$ "Semantic knowledge is the knowledge obtained via the analysis of the meanings in the context."
$D_5 =$ "From the learner's point of view, there are four categories of knowledge, that is, constructive knowledge, procedural knowledge, strategic knowledge and semantic knowledge."

By applying the keyword abstraction algorithm, the set of keywords in each question is obtained as follows. $K(Q_1) = \{$constructive, knowledge$\}$, $K(Q_2) = \{$property, procedural, knowledge$\}$, $K(Q_3) = \{$strategic, knowledge$\}$; $K(Q_4) = \{$semantic, knowledge, learner$\}$, $K(Q_5) = \{$knowledge, learner, category$\}$, $K(D_1) = \{$constructive, knowledge, learner$\}$, $K(D_2) = \{$procedural, knowledge$\}$, $K(D_3) = \{$Strategic, knowledge, activity$\}$, $K(D_4) = \{$Semantic, knowledge, analysis$\}$ and $K(D_5) = \{$learner, category, knowledge, constructive, procedural, strategic, semantic$\}$. In summary, Table 1 shows the 10 keywords symbolized by $K_1, K_2, \ldots, K_{10}$ extracted from the questions and answer documents.

With the symbolizations, the question vectors CVQ and the document vectors CVD can be constructed as shown in Tables 2 and 3. $\mathrm{CVQ}(Q_i, K_j) = 1$ indicates the keyword K_j is contained in the question Q_i, and 0 otherwise. While $\mathrm{CVD}(D_i, K_j)$ is the initial relevance weight between the document D_i and the keyword K_j which is given randomly or determined by the $TF \times IDF$ model. To optimize the document vector weights, the domain-experts provide a training matching matrix $\mathrm{Match}(Q_p, D_q)$ between the documents and the questions. $\mathrm{Match}(Q_p, D_q) = 1$ indicates the document D_q is the correct answer to the question Q_p, and 0 otherwise.

Table 1
Ten keywords abstracted from the questions and answer documents

K_1	Learner
K_2	Knowledge
K_3	Category
K_4	Constructive
K_5	Strategic
K_6	Procedural
K_7	Semantic
K_8	Activity
K_9	Analysis
K_{10}	Property

Table 2
Exemplary question vectors CVQ(Q_i, K_j)

CVQ(Q_i,K_j)	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
Q_1	0	1	0	1	0	0	0	0	0	0
Q_2	0	1	0	0	0	1	0	0	0	1
Q_3	0	1	0	0	1	0	0	0	0	0
Q_4	1	1	0	0	0	0	1	0	0	0
Q_5	1	1	1	0	0	0	0	0	0	0

Table 3
Exemplary document vectors CVD(D_i, K_j)

CVD(D_i,K_j)	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
D_1	0.5	0.23	0	0.5	0	0	0	0	0	0
D_2	0	0.08	0	0	0	0.75	0	0	0	0
D_3	0	0.15	0	0	0.5	0	0	1	0	0
D_4	0	0.23	0	0	0	0	0.5	0	1	0
D_5	0.5	0.38	1	0.5	0.5	0.25	0.5	0	0	0

Table 4
Exemplary question-document matching matrix Match(Q_p,D_q)

Match(Q_p,D_q)	D_1	D_2	D_3	D_4	D_5
Q_1	1	0	0	0	0
Q_2	0	1	0	0	0
Q_3	0	0	1	0	0
Q_4	0	0	0	1	0
Q_5	0	0	0	0	1

In the following, the detailed steps of the proposed genetic approach to optimizing the weights of keywords associated with each document are illustrated.

- Step 1: The domain-experts provide the training question cases and a question-to-document matching matrix. The system automatically abstracts the keywords from the given questions and documents to obtain the question and document vectors. Without loss of generality, these matrices are assumed to be set to those as shown in Tables 2–4.
- Step 2: The genetic approach generates an initial population of five chromosomes as follows. The first chromosome (C_1) is encoded by the keyword weights associated with each document determined using the $TF \times IDF$ model. For example, K_2 appear in D_1 three times and appear total 13 times in all documents. The $TF \times IDF$ model gives the way of determining the value of $\mathrm{CVD}(D_1, K_2)$ as $3/13 = 0.23$. The other weights can be derived similarly. The rest four chromosomes (C_2, C_3, C_4, and C_5) in the population are generated by drawing random weights with values ranging between 0.0 and 1.0. As such the initial population is generated and shown in Table 5.
- Step 3: The genetic approach evaluates the fitness of each chromosome using the fitness function. For example, the fitness of chromosome C_1 is computed as follows. First, the answer document for question Q_1 is derived by calculating $FN(K(Q_1), K(D_i))$ for $i = 1, \ldots, 5$ according to Eq. (2) and obtain their values as 0.73, 0.08, 0.15, 0.23, and 0.88, respectively. Thus, the answer document for Q_1 is D_5 because $FN(K(Q_1), K(D_5))$ is the maximum value among all documents. The answer document for other questions can be derived similarly. Second, the fitness of chromosome C_1 can be derived according to Eq. (3) and is equal to $FN(K(Q_1), K(D_5)) \times \mathrm{Match}(Q_1, D_5) + FN(K(Q_2), K(D_2)) \times \mathrm{Match}(Q_2, D_2) + FN(K(Q_3), K(D_5)) \times \mathrm{Match}(Q_3, D_5) + FN(K(Q_4), K(D_5)) \times \mathrm{Match}(Q_4, D_5) + FN(K(Q_5), K(D_5)) \times \mathrm{Match}\,(Q_5, D_5) = 0 + 0.83 + 0 + 0 + 1.88 = 2.71$. Analogously, the fitness values (f_i) of the other chromosomes can be derived and they are equal to 0.95, 1.311, 0.0, and 0.0, as displayed in Table 6.

Table 5
Initial population

Chromosomes	Doc.	Keywords									
		K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
C_1	D_1	0.5	0.23	0	0.5	0	0	0	0	0	0
	D_2	0	0.08	0	0	0	0.75	0	0	0	0
	D_3	0	0.15	0	0	0.5	0	0	1	0	0
	D_4	0	0.23	0	0	0	0	0.5	0	1	0
	D_5	0.5	0.38	1	0.5	0.5	0.25	0.5	0	0	0
C_2	D_1	0.02	0.05	0.4	0.9	0.55	0.01	0.04	0.07	0.09	0.04
	D_2	0.48	0.23	0.18	0.02	0.01	0.29	0.36	0.34	0.14	0.008
	D_3	0.7	0.14	0.27	0.4	0.025	0.08	0.11	0.12	0.25	0.27
	D_4	0.62	0.51	0.55	0.35	0.091	0.14	0.22	0.25	0.66	0.22
	D_5	0.49	0.32	0.3	0.1	0.04	0.03	0.66	0.47	0.11	0.1
C_3	D_1	0.07	0.6	0.22	0.07	0.07	0.41	0.66	0.54	0.21	0.0112
	D_2	0.52	0.68	0.51	0.8	0.02	0.54	0.11	0.22	0.05	0.091
	D_3	0.19	0.47	0.16	0.25	0.06	0.23	0.22	0.14	0.054	0.014
	D_4	0.02	0.45	0.01	0.47	0.012	0.1	0.36	0.35	0.098	0.054
	D_5	0.3	0.2	0.05	0.04	0.008	0.09	0.14	0.2	0.002	0.035
C_4	D_1	0.22	0.4	0.05	0.06	0.07	0.6	0.14	0.08	0.01	0.08
	D_2	0.31	0.4	0.054	0.25	0.045	0.24	0.25	0.034	0.02	0.06
	D_3	0.58	0.25	0.06	0.14	0.068	0.36	0.36	0.04	0.035	0.014
	D_4	0.04	0.1	0.07	0.5	0.06	0.25	0.46	0.64	0.014	0.06
	D_5	0.02	0.01	0.09	0.1	0.02	0.1	0.22	0.12	0.03	0.04
C_5	D_1	0.22	0.012	0.06	0.06	0.04	0.65	0.23	0.36	0.14	0.001
	D_2	0.68	0.07	0.5	0.07	0.012	0.032	0.14	0.12	0.32	0.11
	D_3	0.026	0.05	0.42	0.65	0.02	0.04	0.35	0.35	0.12	0.68
	D_4	0.01	0.4	0.36	0.14	0.6	0.24	0.33	0.22	0.33	0.54
	D_5	0.02	0.03	0.21	0.65	0.07	0.032	0.41	0.67	0.43	0.78

Table 6
Roulette wheel selection

C_i	f_i	S_i	A_i	q_i	C'_i
C_1	2.71	0.55	0.55	0.3014	C'_1
C_2	0.95	0.19	0.74	0.7665	C'_2
C_3	1.311	0.26	1	0.5833	C'_3
C_4	0	0	1	0.3424	C'_4
C_5	0	0	1	0.1975	C'_5

- Step 4: The roulette wheel selection (see Section 4) is performed to generate the next population. First, the selection probability (S_i) of each chromosome is computed based on the respective fitness value (f_i). Second, the accumulated selection probability (A_i) is derived by $A_i = \sum_{j=1}^{i} S_j$. The next population is obtained by drawing five random real numbers $q_k \in [0.0, 1.0]$, and C_i will be selected if $A_{i-1} < q_k \leqslant A_i$. As for the example, the selection probability, the accumulated selection probability, and the drawn random numbers are tabulated in Table 6. Therefore, the replicated chromosomes (C'_i) are $C'_1 = C_1, C'_2 = C_3, C'_3 = C_2$, $C'_4 = C_1$, and $C'_5 = C_1$, which compose the next population.
- Step 5: The two-point crossover operator is adopted in the enhanced genetic approach. Assume that C'_1 and C'_2 (see Table 7) are chosen to mate each other. The two crossover points are determined at random and they are 12 and 20, respectively. The gene values between the two crossover points from C'_1 and C'_2 are interchanged and two new offspring C_6 and C_7 are obtained as shown in Table 8.
- Step 6: Assume that the 32nd gene of C'_3 (see Table 9) is selected to perform mutation. Its current value 0.51 is replaced by any value, say 0.18, in the feasible range (see Table 10).

Table 7
Parent chromosomes before performing the two-point crossover operation

Chromosomes	Doc.	Keywords									
		K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
C'_1	D_1	0.5	0.23	0	0.5	0	0	0	0	0	0
	D_2	0	0.08	0	0	0	0.75	0	0	0	0
	D_3	0	0.15	0	0	0.5	0	0	1	0	0
	D_4	0	0.23	0	0	0	0	0.5	0	1	0
	D_5	0.5	0.38	1	0.5	0.5	0.25	0.5	0	0	0
C'_2	D_1	0.07	0.6	0.22	0.07	0.07	0.41	0.66	0.54	0.21	0.0112
	D_2	0.52	0.68	0.51	0.8	0.02	0.54	0.11	0.22	0.05	0.091
	D_3	0.19	0.47	0.16	0.25	0.06	0.23	0.22	0.14	0.054	0.014
	D_4	0.02	0.45	0.01	0.47	0.012	0.1	0.36	0.35	0.098	0.054
	D_5	0.3	0.2	0.05	0.04	0.008	0.09	0.14	0.2	0.002	0.035

Table 8
Offspring chromosomes obtained after performing the two-point crossover operation

Chromosomes	Doc.	Keywords									
		K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
C_6	D_1	0.5	0.23	0	0.5	0	0	0	0	0	0
	D_2	0	0.68	0.51	0.8	0.02	0.54	0.11	0.22	0.05	0.091
	D_3	0	0.15	0	0	0.5	0	0	1	0	0
	D_4	0	0.23	0	0	0	0	0.5	0	1	0
	D_5	0.5	0.38	1	0.5	0.5	0.25	0.5	0	0	0
C_7	D_1	0.07	0.6	0.22	0.07	0.07	0.41	0.66	0.54	0.21	0.0112
	D_2	0.52	0.08	0	0	0	0.75	0	0	0	0
	D_3	0.19	0.47	0.16	0.25	0.06	0.23	0.22	0.14	0.054	0.014
	D_4	0.02	0.45	0.01	0.47	0.012	0.1	0.36	0.35	0.098	0.054
	D_5	0.3	0.2	0.05	0.04	0.008	0.09	0.14	0.2	0.002	0.035

Table 9
Chosen chromosome to be mutated

Chromosomes	Doc.	Keywords									
		K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
C'_3	D_1	0.02	0.05	0.4	0.9	0.55	0.01	0.04	0.07	0.09	0.04
	D_2	0.48	0.23	0.18	0.02	0.01	0.29	0.36	0.34	0.14	0.008
	D_3	0.7	0.14	0.27	0.4	0.025	0.08	0.11	0.12	0.25	0.27
	D_4	0.62	0.51	0.55	0.35	0.091	0.14	0.22	0.25	0.66	0.22
	D_5	0.49	0.32	0.3	0.1	0.04	0.03	0.66	0.47	0.11	0.1

Table 10
Chromosome obtained by the mutation operation

Chromosomes	Doc.	Keywords									
		K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
C_8	D_1	0.02	0.05	0.4	0.9	0.55	0.01	0.04	0.07	0.09	0.04
	D_2	0.48	0.23	0.18	0.02	0.01	0.29	0.36	0.34	0.14	0.008
	D_3	0.7	0.14	0.27	0.4	0.025	0.08	0.11	0.12	0.25	0.27
	D_4	0.62	0.18	0.55	0.35	0.091	0.14	0.22	0.25	0.66	0.22
	D_5	0.49	0.32	0.3	0.1	0.04	0.03	0.66	0.47	0.11	0.1

Table 11
Finally obtained optimal document keyword weights

Doc.	Keywords									
	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
D_1	0.29	0.01	0.51	0.25	0.02	0.01	0.55	0.014	0.1	0.1
D_2	0.001	0.3	0.15	0.23	0.005	0.6	0.01	0.22	0.13	0.43
D_3	0.28	0.3	0.11	0.003	0.2	0.5	0.05	0.07	0.3	0.05
D_4	0.21	0.38	0.2	0.31	0.4	0.3	0.34	0.12	0.44	0.05
D_5	0.5	0.5	0.1	0.55	0.05	0.01	0.15	0.05	0.04	0.08

Table 12
The answer documents obtained using the weights by $TF \times IDF$ model and the weights by genetic approach

With the weights by $TF \times IDF$		With the weights by genetic approach	
$FN(K(Q),K(D_1))$	$0.73/1.23 = 0.593$	$FN(K(Q),K(D_1))$	$0.81/1.854 = 0.436$
$FN(K(Q),K(D_2))$	$0.08/0.83 = 0.096$	$FN(K(Q),K(D_2))$	$0.451/2.076 = 0.217$
$FN(K(Q),K(D_3))$	$0.15/1.65 = 0.090$	$FN(K(Q),K(D_3))$	$0.69/1.863 = 0.370$
$FN(K(Q),K(D_4))$	$0.23/1.73 = 0.132$	$FN(K(Q),K(D_4))$	$0.79/2.75 = 0.287$
$FN(K(Q),K(D_5))$	$1.88/3.63 = 0.517$	$FN(K(Q),K(D_5))$	$1.1/2.03 = 0.541$
Answer document	D_1	Answer document	D_5

- Step 7: Repeat Steps 3–6 until the stopping criterion is reached. Assume that the finally obtained optimal document keyword weights are shown in Table 11.

4.3.2. Improving reply accuracy using the optimal weights

The accuracy of the retrieved document can be improved using the optimal weights obtained by the proposed genetic approach. Assume that the user input a question Q that is "From the learner's point of view, what are the four categories into which the knowledge can be classified?" The proposed algorithm abstracts a set of keywords, namely {learner, knowledge, category}, from the question by referring to the keyword database, and the question vector thus becomes (1, 1, 1, 0, 0, 0, 0, 0, 0, 0). If the document keyword weights determined by the $TF \times IDF$ model (see Table 3) are employed, the auto-reply system will return D_1 as the answer document because $FN(K(Q),K(D_1))$ has the maximal value among all documents as shown in Table 12. On the other hand, using the optimal weights (see Table 11) obtained by the proposed genetic approach, the returned document will be D_5. Apparently, D_5 quoted as "From the learner's point of view, there are four categories of knowledge, that is, constructive knowledge, procedural knowledge, strategic knowledge and semantic knowledge." is the best-fit answer document to question Q, and the enhanced genetic approach improves the reply accuracy in this illustrative example.

5. Development of an e-learning system with auto-reply mechanism

Based on the novel approach, an e-learning system with auto-question-answering function has been developed. The system will automatically reply most of the questions submitted by the students with the answers provided by the teachers; therefore, the loading of teachers can be significantly reduced. Moreover, as the e-learning system offers 24-h service with instant feedback, students will not be stuck during their learning process while encountering problems. The e-learning system offers two modes for submitting questions, i.e., e-mail and web page. Fig. 5 shows the user interface of the question-submitting web page. Fig. 6 depicts the answer given by the system.

If no feasible answer can be found in the FAQ database, the system will forward the question to the teacher. A management interface is provided to remind and assist the teacher in answering the question. Once the new answer is available, the system will send it to the student via the same mode of submitting the question. Moreover, the teachers can review all of the questions submitted by the students and the answers replied by the

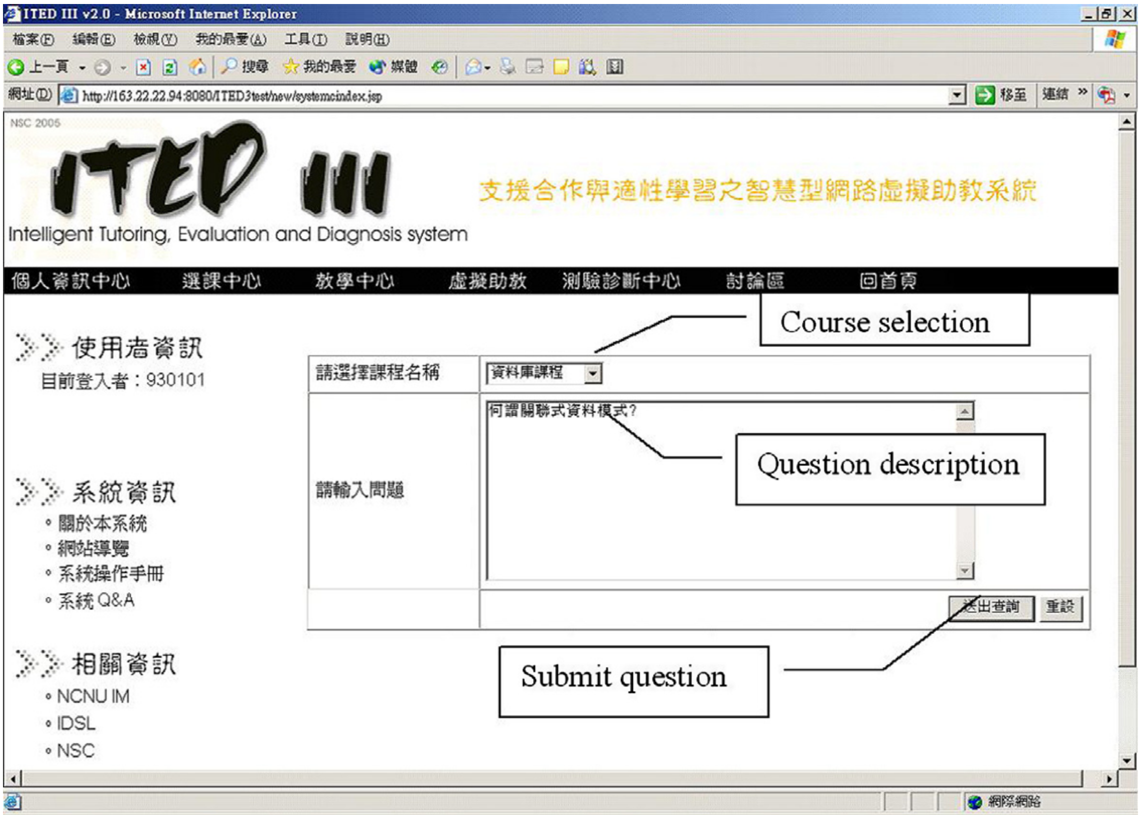

Fig. 5. User interface for submitting questions.

systems with corresponding satisfaction degrees rated by the students, as shown in Fig. 7, which is helpful to the teacher in realizing the learning status of each student and the performance of the system.

6. Experiments and evaluation

In order to evaluate the performance of the proposed auto-reply system, two experiments are conducted. The first experiment tests the system with the domain-experts on two courses and gives the statistical results on the performance measurement. The second experiment applies the system to a course class of 52 students in a university and at the end of this course, in which the participated students are requested to fill out a questionnaire related to the assessment of the system.

6.1. Experiment 1

The developed auto-reply system is experimented with two courses, namely, the "Expert System" and the "Web Programming". The domain-experts are asked to provide the training question cases to optimize the document keyword weights using the proposed genetic approach. Then, the domain-experts provide another set of 30 testing questions which are different from those of the training cases to measure the reply accuracy. Table 13 reports the accuracy results obtained using the auto-reply system with the weights given by $TF \times IDF$ model and the weights determined by the genetic approach. It is observed that the auto-reply system with the weights given by $TF \times IDF$ model can only achieve the reply accuracy of 46% and 40% for the two tested courses, respectively. While the auto-reply system with the optimal weights learned by the genetic approach can improve the accuracy to 93% and 87% which are remarkably higher than the original accuracy levels.

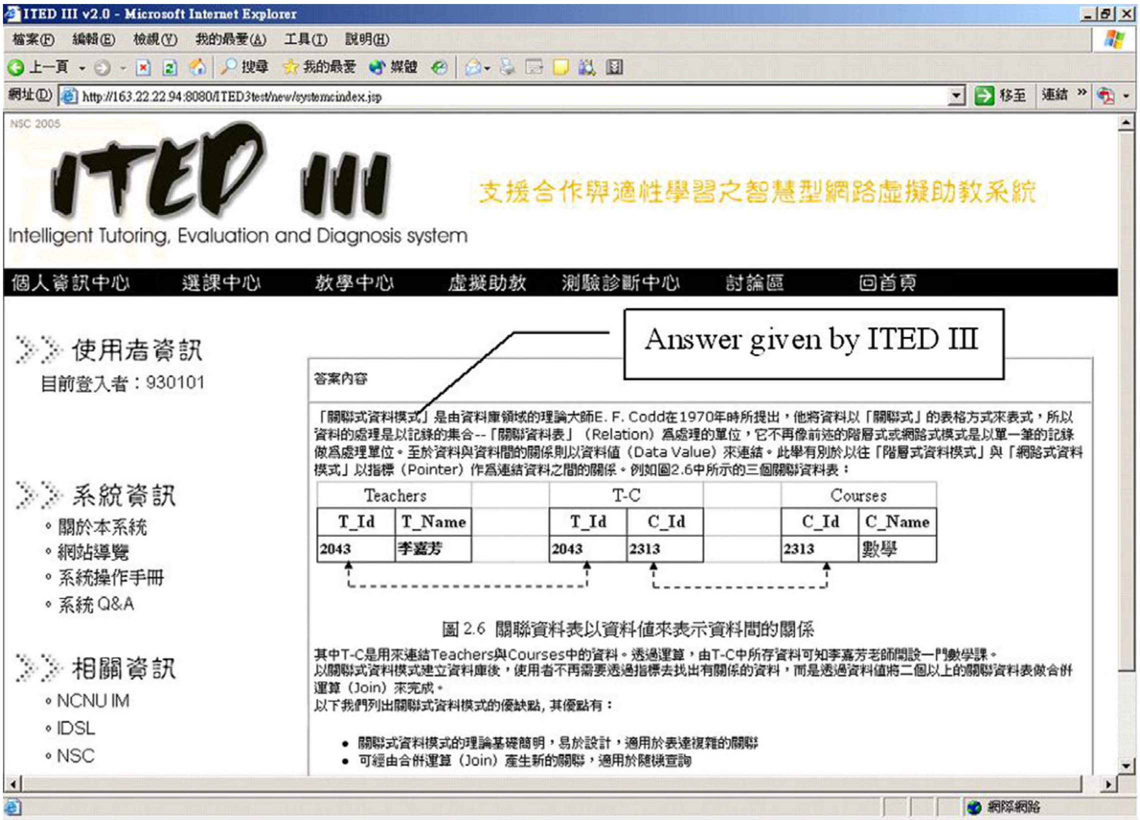

Fig. 6. Illustrative example of the answer given by the system.

6.2. Experiment 2

In this experiment, the optimal weights learned by the genetic approach in Experiment 1 are used to test the ''Web Programming'' course lectured in the sophomore class at the Department of Information Management (http://www.im.ltu.edu.tw/), Ling Tung University. The class consists of 52 students. These students, when needing appropriate on-line documents to explain some course concepts, submit their questions to the auto-reply system. The students are requested to provide feedback on their satisfaction regarding to the answer documents returned by the system. Furthermore, at the end of the course, the students are asked to fill out a questionnaire related to the assessment of the auto-reply system. The details are described as follows.

During the course, there are in total 436 questions submitted to the system by the students. Among them, 208 questions are answered using the $TF \times IDF$ model weights, and the rest are answered by the optimal weights learned using the enhanced genetic approach. The students are requested to provide feedback on their satisfaction regarding the answer document returned by the system. The feedback is constrained in one of the following: ''extremely satisfied'', ''satisfied'', ''fair'', ''unsatisfied'', and ''extremely unsatisfied''.

Table 14 depicts the percentage of feedbacks belonging to each category for the students interacting with the system using the $TF \times IDF$ model weights as well as the weights learned by the innovative approach. When employing the $TF \times IDF$ model, it can be seen that in total 37% of the feedback are belonging to the positive categories of ''extremely satisfied'' and ''satisfied'', 27% are ''fair'', and 36% are either ''unsatisfied'' or ''extremely unsatisfied''.

On the other hand, with the innovative approach, the percentage of feedback belonging to the positive categories of ''extremely satisfied'' and ''satisfied'' increases to 72%, which manifests a significant improvement on the reply satisfaction. The percentage of ''fair'' are 19%, and only 9% of feedback are belonging to the negative categories of ''unsatisfied'' and ''extremely unsatisfied''.

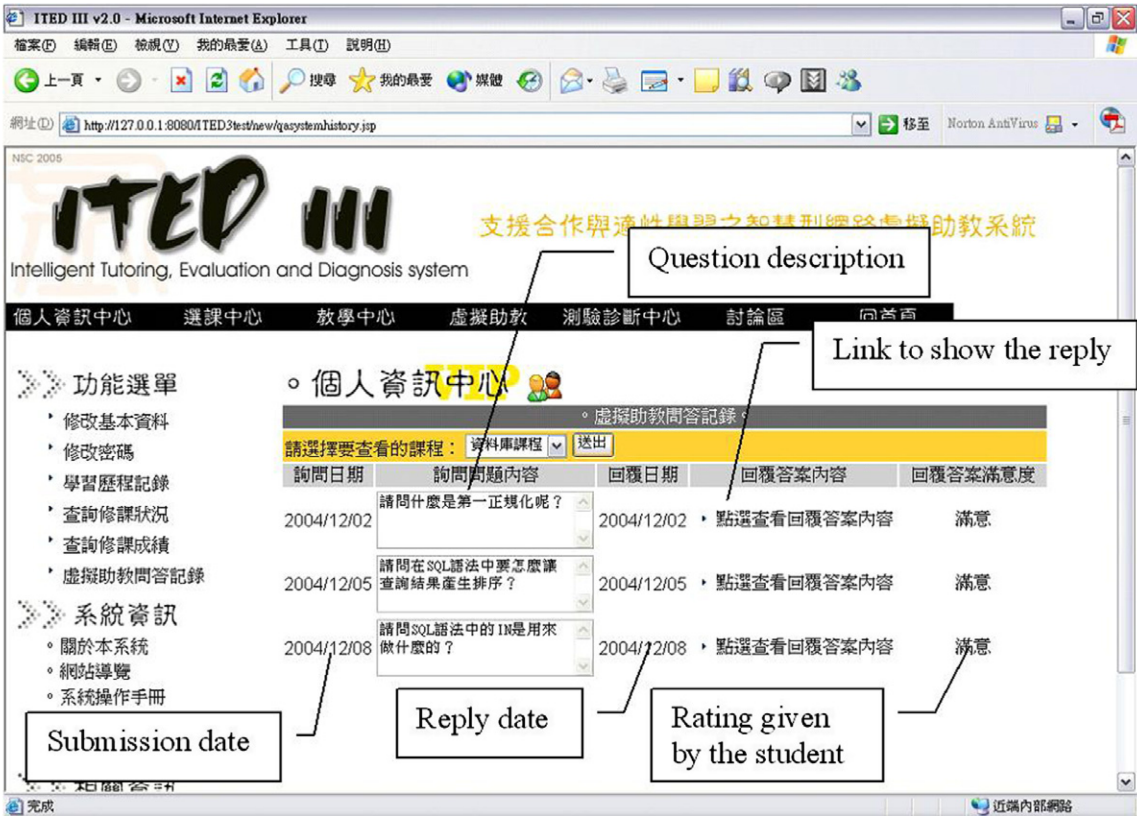

Fig. 7. Management interface for teachers.

Table 13
The reply accuracy obtained by the auto-reply system with the weights by $TF \times IDF$ model and the weights by the genetic approach

Courses	Reply accuracy	
	With the weights by $TF \times IDF$	With the weights by genetic approach
Expert system	7/15 ≒ 46%	14/15 ≒ 93%
Web programming	6/15 ≒ 40%	13/15 ≒ 87%

Table 14
Comparisons of user satisfaction degrees with the two approaches

	Extremely satisfied (%)	Satisfied (%)	Fair (%)	Unsatisfied (%)	Extremely unsatisfied (%)
$TF \times IDF$ model	8	29	27	25	11
Enhanced genetic approach	15	57	19	9	0

6.3. Research limitations

Some findings from the experimental results which impose limitations to this research are summarized as follows. Using the proposed genetic approach, the reply accuracy of the questions for the course of "Expert System" is higher than that for the course of "Web Programming". After a perusal on the submitted questions, the conjecture is that most questions for "Web Programming" are concerned with the programming syntax which is a well-formed description of symbols or words such as "do while" or "for next" instead of a flat description containing several keywords. It is difficult for the keyword abstraction algorithm to identify meaningful keywords from these well-formed descriptions or other mathematical formulas. However, it is advised that the teacher adds more relevant keywords into the database before he/she creates a new course in the auto-reply system such that the reply accuracy could be higher.

7. Conclusions

In an *e*-learning environment, students can learn without being limited by location and time; therefore, it is important to offer full time consultant service. In this study, an e-learning system with auto question-replying mechanism is presented, which can offer student question-answering service 24 h a day, and hence the learning performance of the students can be improved by reducing their discouragements during the learning process. Moreover, an enhanced genetic approach is proposed to adjust the weights of the keywords for each candidate answer, such that the accuracy of the auto-reply mechanism can be improved.

In the near future, several relevant studies are going to be proceeded, including the development of a mobile device version, which is helpful in assisting the students who participate in the course out side the classroom; the applications for enterprise employee training, especially for the high technology companies (e.g., semiconductor designers and producers) who's knowledge advance fast. Moreover, some data analysis methods will be employed to enhance the teacher-facility of the e-learning system; for example, the on-line analytic program technology and the data mining approach can be used to analyze the relationships among the student profiles, portfolios and their feedbacks, which might be helpful to the teacher in maintaining the FAQ database and determining the tutoring strategies to be applied.

Acknowledgements

This study is supported in part by the National Science Council of the Republic of China under Contract Numbers NSC 95-2524-S-024-002 and NSC 95-2520-S-024-003.

References

Adriaa, M., & Chowdhury, S. D. (2004). Centralization as a design consideration for the management of call centers. *Information & Management, 41*, 497–507.

Burnetta, G., & Bonnici, L. (2003). Beyond the FAQ: Explicit and implicit norms in Usenet newsgroups. *Library & Information Science Research, 25*, 333–351.

Cohen, W. (1996). Learning rules that classify E-mail. *AAAI spring symposium on machine learning in information access*.

Cooper, E. (1996). *Improving FAQfinder's performance: Setting parameters by genetic programming*.

Croft, W. B. (1987). Approaches to intelligent information retrieval. *Information Processing and Management, 23*(4), 95–110.

Feng, C. W., Liu, L., & Burns, S. A. (1997). Using genetic algorithms to solve construction time-cost trade-off problem. *Journal of Computing in Civil Engineering, ASCE, 11*(3), 184–189.

Hwang, G. J. (2003). A concept map model for developing intelligent tutoring systems. *Computers & Education, 40*(3), 217–235.

Hwang, G. J., Lin, Bertrand M. T., & Lin, T. L. (2006). An effective approach for test-sheet composition from large-scale item banks. *Computers & Education, 46*(2), 122–139.

Han, K. H., & Kim, J. H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation, 6*(6), 580–593.

Hansen, P., & Jarvelin, K. (2005). Collaborative information retrieval in an information-intensive domain. *Information Processing and Management, 41*, 1101–1119.

Hoch, R. (1994). Using IR techniques for text classification in document analysis. In *17th ACM international conference on research and development in information retrieval* (pp. 31–40).

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.

Jones, L. P., Gassie, E. W., & Radhakrishnan, S. (1990). INDEX: The statistical basis for an automatic conceptual phrase-indexing system. *Journal of American Society for Information Science, 41*(2), 87–98.

Jonassen, D. H. (2000). *Computers as mindtools for schools*. Upper Saddle River, NJ: Merrill.

Kerenidis, I., & de Wolf, R. (2004). Quantum symmetrically-private information retrieval. *Information Processing Letters, 90*, 109–114.

Krulwich, B. (1995). Learning document category descriptions through the extraction of semantically significant phrase. *IJCAI-1995 workshop on data engineering for inductive learning*, Montreal, Canada, August 20.

Larkey, L. S., & Connell, M. E. (2005). Structured queries, language modeling, and relevance modeling in cross-language information retrieval. *Information Processing and Management, 41*, 457–473.

Li, M. Y., & Tseng, Judy C. R. (2001). Development of an intelligent network-based customer service system. In *2001 Taiwan: Academic network conference (TANET2001)*, October 24–26, 2001.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Genetic algorithms for flowshop scheduling problems. *Computers and Industrial Engineering Journal, 30*(4), 1061–1071.

Rau, P. L. Patrick, Chen, S. H., & Chin, Y. T. (2004). Developing web annotation tools for learners and instructors. *Interacting with Computers, 16*, 163–181.

Salton, G., & Buckley, C. (1998). Term weighting approaches in automatic information retrieval. *Journal of Information Proceeding and Management, 24*(3), 513–524.

Salton, G., & McGill, M. J. (1983). *Introduction to modern informational retrieval*. NY: McGraw-Hill.

Savoy, J. (1996). An extended vector-processing scheme for searching information in hypertext systems. *Information Processing & Management, 32*(2), 155–170.

Spink, A. (1995). Term relevance feedback and mediated database search: Implications for information retrieval practice and system design. *Information Processing & Management, 31*(2), 161–171.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the third international conference on genetic algorithms and their applications* (pp. 2–9). San Mateo, CA: Morgan Kaufmann.

Syu, I., & Lang, S. D. (2000). Adapting a diagnostic problem-solving model to information retrieval. *Information Processing and Management, 36*, 313–330.

Toffolo, A., & Lazzaretto, A. (2002). Evolutionary algorithms for multi-objective energetic and economic optimization in thermal system design. *Energy, 27*, 549–567.

Tseng, Judy C. R., & Hwang, G. J. (2004). A novel approach to diagnosing student learning problems in e-learning environments. *WSEAS Transactions on Information Science and Applications, 1*(5), 1295–1300.

Tseng, Judy C. R., & Hwang, G. J. (2006). Development of an automatic customer service system on the internet. *Electronic commerce research and applications*, doi:10.1016/j.elerap.2006.04.009.

Vakkari, P. (1999). Task complexity, problem structure and information actions integrating studies on information seeking and retrieval. *Information Processing and Management, 35*, 819–837.

Witt, L. A., Andrews, M. C., & Carlson, D. S. (2004). When conscientiousness isn't enough: Emotional exhaustion and performance among call center customer service representatives. *Journal of Management, 30*(1), 149–160.

Yuwono, B., & Lee, D. L. (1996). WISE: A world wide web resource database system. *IEEE Transactions on knowledge and Data Engineering, 8*(4), 548–554.