A Modification on Q&A systems based on CBR and GA

NAMITA MAHIN
Faculty of Engineering
Islamic Azad Univresity
Poonak,Tehran
IRAN

ALI MOEINI
Engineering and Science Department
Faculty of Engineering
University of Tehran
IRAN

*Abstrac*t: "Question and Answering" (Q&A) is a task to obtain appropriate answers for given domain independent questions written in natural language from a large document collection [1].
Due to limitations of IR models to recall the textual documents, case-based reasoning (CBR) techniques can be applied on document retrieval .In existed question and answering engines the feature weights are maintained gradually according to the users response from the beginning and so the prediction is too limited and maintaining process is boring and not efficient. In this paper In order to improve the prediction accuracy a new approach based on genetic algorithm is proposed.

Key-words: Q&A, Case-based reasoning ,Genetic algorithm

1 Introduction

Traditionally, we got information and instruction from school education or books, newspapers, radio, television. Although they are easy to be obtained and easy to be understood, the following Internet and web based occurrence still caused huge impacts of learning in our life .

Therefore, in recent years, more and more researches had put many efforts to find methods of building intelligent e-Learning systems to assist learners to learn easier in convince environments [18] .

With deployment of Internet, E-learning and virtual learning provide a promising way in education. In learning process, student needs to get experts instructions online which can let them feel learning with teachers face to face, just the same as a real class room. Q&A system mediates interactions between an expert and a question-asking user. they use their experience referring questions to expert users to answer new questions by retrieving previously answered ones [2].

The main function is to analysis the submitted question automatically and find the probably answer to the users. The current Q&A systems are based on mail-solution, keyword-matching or word segmentation techniques [9, 10]. These systems can deal with the submitted questions in a sense. However with the growth of the number of users and the questions, the process cycle will become longer and the matching accuracy will become lower due to different presentation of the question and changing interest of users [3].

When trying to recall relevant textual documents, usually Information Retrievals (IR) techniques are applied. The systems which use these techniques help users to find some related results in response to their needs expressed as queries, from large amount of data collections. The user expresses his/her need in a natural language description, and such systems will then mine through information space to find the related texts with respect to the query as beneficial as possible [17].

A major limitation of IR model is the knowledge such as domain-specific term and the relationship among these can hardly be utilized when searching the document collection .A pure keyword-based search on the other hand is not powerful enough to provide a flexible question answering system .For example virtually any natural language text can be paraphrased such that nearly every keyword is changed but the semantics of the entire expression is still very close to the original text.

Another limitation of IR techniques is that they have problem in dealing with semi-structured documents, i.e document for which a certain structure can be assumed [4].

Case Based Reasoning (CBR) systems solve new problems by re-using the solutions to similar, previously solved problems. The main knowledge

source for a CBR system is a database of previously solved problems and their solutions; the *case knowledg* [5] .In a problem situation the key idea is to look for similar problem descriptions from the past and to utilize the solution that works for this past problem . In CBR terminology, a *case* usually denotes a *problem situation*. A previously experienced situation, which has been captured and learned in a way that it can be reused in the solving of future problems, is referred to as a past case, previous case, stored case, or retained case.Correspondingly, a new case or unsolved case is the description of a new problem to be solved.
case-based reasoning is – in effect – a cyclic and integrated process of solving a problem, learning from this experience, solving a new problem, etc [6]. Based on above advantages, CBR was applied to traditional Q&A systems by Penghan [11]. Basically system makes inferences using analogy to obtain similar experience for solving problems.Similarity measurements between pairs of features play central role in CBR. Several approaches have been presented to improve the case retrieval effectiveness .These include the parallel approach [12],instance-based learning algorithm [13],fuzzy logic method [14],neural network meth].
In this paper based on the architecture introduced by Yonggang Fu and Ruimin Shen [3], we used different GA methods to optimize the feature weighting. In section 2 their architecture is presented. In section 3 GA approach is defined. Then in section 4 the experimental results of our approach are presented. Finally in section 5 the conclusion is given.

2 Architecture of the Auto Q & A System

this system based on CBR technique, is divided into two separate modules , the first one called Case Authoring Module and the second one Automatic Q & A Engine.The Case Authoring Module is to represent the unstructured field knowledge structurally based on empirical expert know edge and application background. All of these structural representations can be transferred into the question answer instances and stored in the system case repository . The Automatic Q & A Engine is the kernel of the system . It is triggered by the keywords or description of the problem and returns the ranked similar problems related to the description according to the scores . So the user can select the most similar problem and get the answer in details. Furthermore , the system provides a feedback module for the users score.The architecture of the Q &A system is as shown in Fig1.

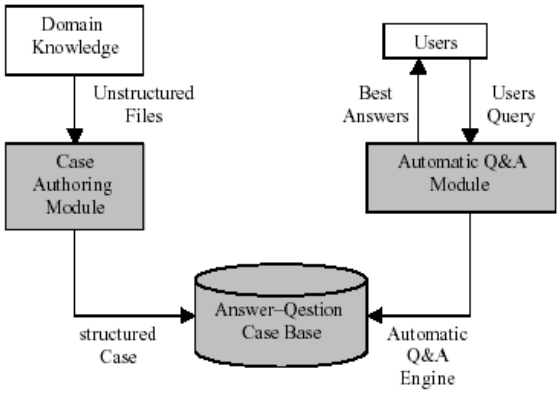

Fig. 1. The syestem architecure

All the questions , answers and the relativity of them are accessed thorough the standard web interfaces. The users , especially the students produced a great number of questions and potential answers during the learning process . All of the questions and answers are assembled in log files.the index architecture of the relationship between questions and answers were trained based on the log files . This process is running during the life cycle of the system , which makes the Q & A system become a closed – loop system . Case retrieval searches the case base to select existing cases sharing significant features with the new case.

Nearest – neighbor matching is a quite direct method that uses a numerical function to compute the degree of similarity.A typical numerical function is shown in the following formula [16]:

$$\sum_{i=1}^{n} W_i \; sim\left(f_i^I, f_i^R\right)$$

Where W_i is the weight of the *i*th feature and $\Sigma^n_{i=1} W_i = 1$, f_i^I is the value of the *i*th feature for the input case , f_i^R is the value of the *i*th feature for the retrieved case , and sim ( ) is the similarity function for, f_i^I and f_i^R .

How to decide feature weights in a case base in a multi-user environment become a key problem . Penghan have presented a model to dynamically adjust the feature weights [11] . Since it changes weights gradually with the users reaction , convergence rate of the feature weight may be very slow and boring . in this architecture composition of three major processes is proposed to maintain the feature weights . One case base includes both training cases and old case (Fig. 2). The similarity process computes the similarity between an input train case and an old case. The similarity (named overall similarity Degree) is derived by summing each degree of similarity resulting from comparing each pair of corresponding case features out of the selected training case and old case . The OSD is expressed as following :

$$OSD = \sum_{i=1}^{n} W_i S_{i,j,k}^{ei}$$

Where I=1,2 n,n is the total number of features in a case , W_i is the weight of *i*th feature . This feature is generated from the weighting process by the GA. e_i represent the power of $S_{i,j,k}$, which represents the degree of similarity for the *i*th feature between the training case$_j$ and the old case$_k$ (j=1 to p;k=1 to q) . $S_{i,j,k}$ is used as an index to describe the similarity level for certain case features for one training case against that of one old case .This can be expressed as follows:

$$S_{i,j,k} = 1 - \left| Feature_{Casej} - Feature_{Cosek} \right| \Big/ Range_i$$

Where Range$_i$ is the longest distance between two extreme values for the *i*th feature [3].

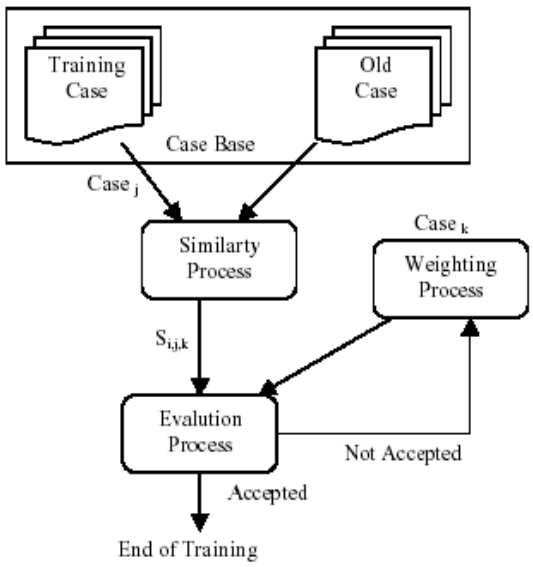

Fig. 2. Training architecture

3 Genetic Algorithm Approach

Genetic algorithms (GAs) are randomized parallel search algorithm that search from a population of points. GAs are well suitable to the high dimensionality of the search space and the combination of the crossover operator and selection preserve successful groups of feature weights [7].

Genetic algorithms have received considerable attention regarding their potential as a novel optimization problem. There are three major advantages when applying genetic algorithm to optimization problems:

1- Genetic algorithm do not have much mathematical requirement about the optimization problems. Due to their evolutionary nature genetic algorithm will search for solution without regard to specific inner working of the problem .genetic algorithm can handle any kind of functions.

2- The ergodicity of evolution operators make genetic algorithm very effective at performing global search (in probability).

3- Genetic algorithms provide us a grate flexibility to hybridize whit domain dependent heuristic to make an efficient implementation for a specific problem [8].

In genetic algorithm application major concerns are genome representation, initialization, selection,

cross over and mutation operations, stopping criteria and the most important the fitness function.

In the system which is introduced by Yonggang Fu and Ruimin Shen [3] the fitness value is defined as the number of old cases whose solution match the input case solution, i.e the training case . OSD has been used for assessing the similarity between the input case and the old case. For each the similarity process batch executed with specific input case, $case_j$, then q OSDs are produced since q old cases have been compared with the input $case_j$. With higher OSD, the matching between the retrieved old case and the input case increases. The purpose for introducing the GA is to determine the most appropriate set of weight values that can direct a more effective search for higher OSDs to match the input case .To determine which whose outcome feature can be adopted as the outcome feature for the input case they proposed that the top 10% OSDs in those old case are used to represent the final solution for each batch of the similarity process execution for a given training $case_j$. The derived final expected outcome feature is denoted as O'_j as opposed to the real outcome feature O_j for a given training $case_j$. Weighting process is applied to minimize the overall difference between the original real outcome feature and the expected outcome features. The fitness function is as follow:

$$Maximize Y = \sum_{j=1}^{p} y_i$$

where P is the total number of training cases ;y_i is the matched result between the expected outcome and the real outcome, i.e if $O'_j=O_j$ then y_i is 1 ; otherwise is 0.

In the selection process, Roulette Wheel method and the single point crossover with moderate crossover probability have been used. First a big mutation rate was adopted. Then a simulated annealing method was used to decrease the mutation rate to zero.

In this paper we tried to compare different methods of GA to optimize the feature weights.
There are only two kinds of operations in genetic algorithm:

- genetic operation: crossover and mutation
- evolution operation : selection

The genetic operations mimic the process heredity of genes to create new offspring at each generation. The evolution operation mimics the process of Darwinian evolution to create populations from generation to generation. Crossover is the main genetic operator. A higher crossover rate allows exploration of more of the solution space and reduces the chances of settling for a false optimum ; but if this rate is too high , it results in the wastage of a lot of computation time in exploring unpromising regions of the solution space.
Mutation is a background operator which produces spontaneous random changes in various chromosomes. If the mutation rate is too low, many genes that would have been useful are never tried out; but if it is to high, there will be much random perturbation, the offspring will start losing their resemblance to the parents and the algorithm will lose the ability to learn from the history of the search [8].

4 The modification and experimental results

In this paper the Q&A system is implemented based on a collection of questions and answers .The features and their values are extracted by SMART program .We have converted this information into the case bases which are maintained in the tables. Rows of these tables are the cases and the columns are the features. For running GA the population size is 20 and the length of each chromosome is 140 bit which are initialized randomly in binary code. The number of old cases is 50 and 20 cases are used as the training cases. Similarity between each training case and old cases are calculated through the OSD algorithm which mentioned earlier. Maximum similarity between each training case and old cases are kept. The desired maximum based on above happens when all feature values in training cases and old cases are the same. According to the OSD formula, the maximum OSD is equal to the number of features. So our fitness function could be defined as follows:

$$y = 1 - \left| ftr\_n - \max\_old\_train \right| \Big/ ftr\_n$$

$$Fitness = \sum_{i=1}^{t} y$$

Where t is the number of train cases, ftr_n is the number of features and max_old_train is the maximum similarity between each train case and old cases according to the chromosomes which contain the features weights.

We have two stopping criteria: first the maximum number of generation which is 10 generations and second is the minimum error which defined as the difference between the maximum outcome fitness and the desired fitness. According to the fitness formula defined above, the value of this desired fitness is equal to the number of train cases.

The best weights for features are the values of the chromosome which results the best fitness at the end of GA run.

The data set has been tested in four situations with various conditions as shown in Table 1.
In this table #Crp is the number of cross points, PC is the crossover rate and selection is the method used for selection .Mutation rate in all situations is 0.1 at first and then increased to zero by using simulated annealing method.

Table 1. Dataset Testing Conditions

	#Crp	PC	Selection
Situation1	2	0.5	15chrom with higher fitness and 5 with lower
Situation2	2	0.5	Roulette Wheel
Situation3	1	0.5	Roulette Wheel
Situation4	2	0.8	Chrom with higher fitness

The result of the program and the comparison with simulated previous system is depicted in Fig.3. From this figure it is shown that in most cases the minimum error in our system is less than the previous system and also the best result occurred when:

- The number of cross points and the crossover rate increase.
- Chromosomes with higher fitness values are selected.

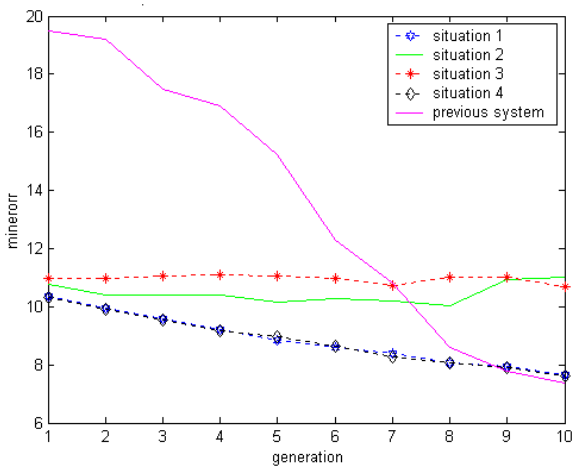

Fig.3 Comparison of Situations and previous system.

5 Conclusion

Defining appropriate feature weighting value is a crucial issue for effective case retrieval in Q&A systems. In this paper a GA plus CBR approach based on the F.Yonggang, & S.Ruimin system was used to determine the fittest weighing values for improving the case identification accuracy. Their system had shown significant promise for Q&A system accuracy and efficiency compared to non GA models. We tried to optimize weighting by working on GA operators and comparing them together.

References

[1] D. Kawahara,S. Kurohashi & N. Kaji .Question and Answering System based on predicate- Argument Matching.

[2] J.Budzik, & K.Hammond . Q&A: A system for the Capture, Organization and Reuse of Expertise. *In Proceedings of the Sixty-second Annual Meeting of the American Society for Information Scienc,1999.*

[3] F.Yonggang, & S.Ruimin . GA based approach in Q&A syetem ,2003.

[4] M.Lenz, A.Hubner, & M.Kunze. Qestion Answering with Textual CBR.

[5] S.Craw . Introspective Learning to Build Case-BasedReasoning (CBR) Knowledge Containers.

[6] A. Aamodt, & E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications.IOS Press, Vol. 7: 1, pp. 39-59*,1994.

[7] J.Jarmulak, S.craw & R. Rowe . Self-Optimising CBR Retrieval. *Proceedings 12th IEEE International Conference on Tools with Artificial Intelligence, pp. 376–383*, 2000.

[8] M.Gen & R.Cheng . *Genetic Algorithms And Engineering Design* . Wiely , 1997.

[9] S.Barry, T.k.Mark, & C.Padraig. Hierarchical case-base reasoning integrating case-base and decompositional problem-solving techniques for plan control software design. *IEEE Transaction on knowledge and Data Engineering , 13(5),793 812,*2001.

[10] K.Racine, & Q.Yang. Maintaining Unstructured case base. *Proceeding of the Second International Conference on Case-Based Reasoning, ICCBR-97, Providence RI, USA, pp.553 564*, 1997.

[11] Penghan, S.Ruimin, & Y.Fan. Intelligent Q&A system based on case based reasoning .*15th Australian Joint Conference on Artificial Intelligence* ,2002.

[12] J.l.Koldner. Retrieving events from a case memory .*Proceedings of the Case-based Reasoning Workshop ,San Mateo,CA:Morgan Kaufmann,*1988.

[13] D.W.Aha. Tolerating noisy, irrelevant and novel attribute in instanced-based learning algorithm. *International Journal of Man Machine Studies,36,267,287,*1992.

[14] B.C.Jeng , & T.P.Liang. Fuzzy indexing and retrieval in case-based systems. *Expert Systems with Application ,8,135 142* ,1995.

[15] Z.Zhong & Y.Qiang. Feature weight maintenance in case bases using introspective learning. *Journal of Intelligent Information Systems,16,95 116,*2001.

[16] J.L.Kolodner . Case-based reasoning. San Mateo, CA:Morgan Kaufmann,1993.

[17] S.Karimi .A.Moeini & M.Hejazi. Information Mining Based on Fusing Results of Multi_perspective cluster-based Summarization. *To be Appeared in WSEAS Transaction on computer Issues 4,volume 3*,October 2004.

[18] C.Y.CHEN & W.S. LO. An Agent E-Learning System for Interactive and Collaborative Communication. *to be Appeared in WSEAS Transaction on computer Issues 4* , 2004.