Designing Question Answering System by Using the Genetic Algorithms

Mohsen Shakibafakhr, Mojtaba Khayat
Sama Technical and Vocational Training College, Islamic Azad University, Shoushtar Branch, Shoushtar, Iran
Sama Technical and Vocational Training College, Islamic Azad University, Shoushtar Branch, Shoushtar, Iran
shakibafakhr@yahoo.com

ABSTRACT

Question Answering (QA) systems made possible to ask questions and retrieve answers using natural language (NL) queries, rather than the keyword-based retrieval mechanisms used by current search engines. Currently, there exists several QA approaches, and, according to recent evaluation results, most of them are complementary. Most QA systems combine information retrieval with extraction techniques to identify a set of likely candidates and then utilize some ranking strategy to generate the final answers. In this study, we presented the results of design, implementation and evaluation of a restricted domain Question Answering System, Based on the knowledge base. Here, we have designed a question answering system, by using the genetic algorithms, that its target is obtaining the most relevant answers in the least possible time. In this system, is used the genetic algorithm, to search in the knowledge base. The knowledge base is made from web pages. To validate the proposed system, we use the genetic algorithm method, and optimized genetic algorithm method. After the experiments, the results of optimized genetic algorithm method has been significantly improved, compared with genetic algorithm, and responds more quickly to questions.

Keywords: Question Answering System, Genetic Algorithm, Mutation, Crossover, Information Retrieval.

1. INTRODUCTION

Information retrieval Systems (IR), will retrieve all documents are relevant to the user query from the existing resources in our knowledge base by taking a few keyword from the user, in the limited time. Most of documents that are retrieved by search engines in this method are associated with the user queries only in terms of lexical form, and not semantic. These engines instead of referencing the exact answer to users' questions will return the documents that are relevant to user’s questions (Shamsfard and Yarmohammadi, 2010).

These systems have some major problems. First, those users have question, but instead the question, some keywords should be entered. On the other hand, usually users have a problem to convert the question to the appropriate keywords, and this conversion, requires the skill that must be achieved over time. In addition, several keywords cannot make the user purpose, that this issue, sometimes impossible to make this conversion.

Due to the users need to accurate and focused information, and to prevent users wasting time, IR systems cannot resolve users' demands (Li and Roth, 2002). So, a new kind of IR, named Question Answering (QA) systems appeared from the late 1970's and early 1980's. In these systems, the user ask his/her natural language question with no restriction in its syntax or semantic. The system is responsible for finding the exact, short, and complete answer at the shortest possible time (Tapeh and Rahgozar, 2008). Question Answering systems divided into two general categories: open domain and restricted domain Question Answering System.
Restricted domain systems are in the specified data ranges, then only able respond to questions in the same range. But the scope of open domain system is infinite, and included almost all types of questions and able to interact with documents of any domain information.

Another division for QA systems is based on the number of languages accepted by these systems. Monolingual systems, receive the question, and respond to it, only with one language. Another group that is called multilingual systems has ability for understand and respond to questions that include several different languages (Baayen et al., 2008).

II. RELATED WORKS

QA systems which are based on searching among a set of documents are usually composed of three main modules (Valero et al., 2010): (1) Question analysis and extension (2) document retrieval (3) answer extraction. The first module analyses the user question to extract the type of question and the expected type of answer or extends it to be used by the next modules. The second module relates to retrieving relevant documents to the user query. It can be replaced by a search engine. The third module extracts the final answer from the documents retrieved by the second module.

All question answering systems, have three steps above, but different methods are used to implement this process.

The first devices for access to the information were textual information retrieval systems that despite the simple are useful, and very widely used. An example of this systems, are Google, AltaVista and MSN Search, where used to find relevant documents on the internet. Some of information retrieval systems are designed for use in textual collection, out of the Internet, such as the SMART (Salton, 1971) and PRISE (Dimmick et al., 1998).

Web Question Answering System, is another example of question answering systems that used the genetic algorithm for ranking. In this system, at first, words will be sent to the Web, and sentences that include the answer, are retrieved. Set of retrieved sentences, are matched with known previous paragraphs, so new answers to be extracted. Therefore, matching procedure is an important parameter. Two strategies based on genetic algorithm, is proposed to improve the matching:

a) GASCA, Trained of Syntactic patterns, derived of (sentence, answers) pairs. For matching, and find alignment, block of words to be translated into near blocks of zeros and ones. Then, according to their fitness, new blocks to be made by the "mutation" and "crossover" operators, that make more probability of matching between training patterns with query.

b) PreGA, Use the semantic relation, to matching the query and training patterns. The previous strategy is based mainly on syntactical patterns, then, if there is not enough syntactical evidence to properly align contextual patterns, the answer will not be unambiguously identified. But with using of this strategy, improved matching between the educational patterns and query (Figueroa and Neumann, 2008).

The basic algorithm that used in this system is shown in Figure1 (Figueroa and Neumann, 2008).
Algorithm GA_QA

input: num_iter, pop_size, N, Q
begin
  Rnd[1] ß create initial population(1, pop_size);
  Evaluate population (rnd[1]);
  Store ((max fit), loc(max fit), db(max fit));
  for i=1: Num_iter
    CAC ß Crossover (rnd[i], pc);
    MAC ß Mutate (rnd[i], pm);
    Rnd[i+1] ß selectPopulation(rnd[i], CAC, MAC);
    Evaluate population (rnd[i+1]);
    Store ((max fit), loc(max fit), db(max fit));
  end
Return db(max (max_fit));
end

Figure 1. GA_QA Algorithm

In this paper, reviews the Question Answering System, called OGA_QA. This system can answer the questions about the inventors. The system knowledge base is made of the web pages. Then, by using an Optimized Genetic Algorithm, select the best sentence in knowledge base, as the answer. The advantage of our proposed system in comparison with Genetic Question Answering Systems is high accuracy to answer the questions.

III. OGA_QA SYSTEM ARCHITECTURE

This section introduces the structure and function of OGA_QA System. This system has implemented, as a restricted domain Question Answering System. Such as most similar systems, this system is composed of three main components too. The main components of this system are: "Sentence Analysis System", "Retrieval and Extraction Answer System" and "Ranking System". The overall structure of system can be seen in Figure 2.

Figure 2. OGA_QA System
The ranking systems (OGA_Ranking), is composed of several components. This system is shown in Figure 3. In this section, by using an Optimized Genetic Algorithm, candidate sentences are ranked, and the highest rank, is placed in the output, as an answer.

A. User Interface

In this section, the system receives question in natural language, along with pop_size and Num_iter.

Num_iter, is the number of mutation. Whatever this number is greater, hence the number of mutations to be great, and thus, is more probability to find answers. But, if this number was too high, reduce the speed of program implementation.

pop_size, is the initial population size. Considering that, the selected population, are chosen in the quad categories for mutations, this number must be multiple of four. If the initial population size was much more, the probability of finding the best sentence, before the mutations, is added. But if its value becomes too high, reduce the speed of program implementation.

B. Sentence Standardization

In this section, the user question is inserted, from the previous stage. Then standardization function applies on it. Standardization is three stages.

First step: check all words in question, and the words that written in capital letters, are converted to the lowercase.

Second step: all extra words (eg. am, and, or, if, is, a, as, an, to, for, the) are removed from the question.
**Third stage:** all the words in question, are examined to find the words that terminated with the (’s, es, er, ional, ion, ors, ive, ions, ed, or, ing), and removed these, from word endings.

**C. Sentence Processing**

In this section, the previous stage output that contains the question keywords, is processed, to detected questions type, so, based on question type, can also predicted the type of answer.

**D. Retrieval system**

In this section, for initial evaluation, some of sentences in the knowledge base are randomly extracted. Evaluation of selected sentences will be discussed in the ranking system.

**E. Ranking System**

The selected sentences that extracted from the knowledge base are ranking. Candidate sentences ranking, is based on question keywords matching with selected sentences and questions types matching with selected sentences type. This system is composed of five stages: Initial evaluation, Sorting, Mutation, Second evaluation, Return answer.

- **Initial evaluation:** At this point, two fitness for each sentence in knowledge base, are used «fit1, fit2». `fit1` is the matching result of all words in question, with any words in all knowledge base, and `fit2`, is for reviews the question type matching with the sentences type of knowledge base. After calculating these value for all selected sentences, total fitness for `i`th sentence in the knowledge base, is calculated by equation 1.

\[
Global\ fitness(i) = (Fit1(i)*W1) + (Fit2(i)*W2)
\]

W1 and W2, are respectively the coefficient of `fit1` and `fit2`, and they values are respectively 0.4 and 0.6.

- **Sorting:** This step includes the sorting `Global fitness` and `initial population` array, with ascending order. Sorting is done by the bubble sort.

- **Mutation and Crossover:** The mutation and crossover operation is done by optimized genetic algorithm. If the sentences have the fitness be over 1.1, one step mutation, if the fitness is between 0.4 to 1.0, four step mutations, and if the fitness is between 0 to 0.3, we have six steps mutations. Here, the "Backward Mutations" and "Random Mutations" is used. In Figure4, the proposed algorithm, named as `OGA_QA` is presented, which used to improve the initial population.
Algorithm 1: OGAQA

\[\text{Algorithm 1: OGAQA}\]

\begin{verbatim}
input: NIT, P, Q
begin
    t = 1; BestFound = 0 ;
    pop[1] = create Initial Population(I, P);
    do
        Evaluate Population (pop[t]);
        Store (BestFound[t]);
        CAC = Crossover (pop[i],pc);
        MAC = Mutate (pop[i], pm);
        pop[t+1] = selectPopulation(pop[t], CAC, MAC);
        t++;
    while (t < NIT)
    return BestFound
end
\end{verbatim}

NIT) Number Of Iteration
P) Population Size
Q) Question
pop) Population

Figure 4. OGA_QA Algorithm

- **Second evaluation**: On the new population that generated from the previous stage the evaluation will be done, same as the first stage. Then, all the above steps, from sorting step, are repeated in \textit{num\textunderscore iter} times.

- **Return answer**: After the mutation was performed on the population in \textit{num\textunderscore iter} times, a sentence that has the highest fitness value is referred as the output of ranking system.

**F. Display the final answer**

This section is used to prevent display the incorrect answer. Note that, sentences that are not contain the correct answer, can have the fitness equal to one, and will prevent from display them in output. So that, if the highest fitness, is less than 1.1, \texttt{<NOT FIND>} message is written in the output. Otherwise, one sentence that has the highest fitness is displayed as the output in the user interface.

**IV. EXPERIMENTAL RESULT**

This section examines the results of different experiments that performed on the proposed method, faced with variety data. Also, the proposed optimized genetic algorithm, are compared to primary genetic algorithm.

First, user question and initial population size and the number of mutation operators, entered into system, then, candidate sentences are scoring, and finally, a sentence with highest score, are
displayed as response to the user. In this paper, two scoring methods that use the genetic algorithm and optimized genetic algorithm are compared with each other.

In chart1, the fitness of OGA_QA system, with initial population equal to 4 and number of implementing mutation, equal to 10, is displayed.

![Chart.1. Fitness of OGA_QA System](image1.png)

As can be seen in Chart1 in the first stage, the fitness of system is equal to 0, then, after several stage of mutation, the fitness equal to 1.8. Considering that this value is greater than 1.1, so, the system is found the answer.

In Chart2, the fitness percentage of GA_QA and OGA_QA system, with the initial population equal to 12 and implementing stages 1 to 10, are displayed.

![Chart.2. Compare the fitness percentage of GA_QA and OGA_QA system with the initial population equal to 12](image2.png)

In chart2, in fifth stage, the fitness average of OGA_QA system, and thus the accuracy of this system is 70% that 42% improved, in comparison with the GA_QA fitness.

In chart3, fitness percent average of GA_QA and OGA_QA system, with implementing stages 1 to 10, are displayed.
Chart 3. Compare the fitness percentage average of GA_QA and OGA_QA system than mutation implementing stages

In this Chart, we see the fitness percentage of the proposed system, in all stages is higher than the previous system.

V. CONCLUSIONS

In this paper, we presented a restricted domain question answering system based on the knowledge base, which uses the optimized genetic algorithm method for ranking. The system knowledge base is composed of the structured texts. To making this knowledge base, used the non-structured web pages. Processors and standardized components of this system, converts the natural language question to the keywords by using these keywords, are scoring the sentences in the knowledge base. Scoring is based on the question keywords matching, with the sentences in the knowledge base, and also, the question type matching with the knowledge base sentences type. After scoring the sentences in the knowledge base, the highest ranking sentence will be displayed in the user interface. According to evaluations, the overall average accuracy of proposed system, have been significantly improved in comparison with GA_QA.

ACKNOWLEDGEMENT

This paper derived from research project "Designing Question Answering System by Using the Genetic Algorithms" that it is done in Sama Technical and Vocational Training College Islamic Azad University, Shoushtar Branch, Shoushtar, Iran. We thank from Sasan Saqaeeyan and Mohammad Behrouzian Nejad for their contributions to the project. We also thank Behnam Kiaemehr and Nayereh fathalipour for their helpful comments.

REFERENCES


Mohsen Shakiba Fakhr is a lecturer at the Shoushtar University; He received his M.Sc. degree in Computer Engineering (Computer Architecture) from Islamic Azad University, Iran, in 2012. He received his B.Sc. in Computer Engineering (Hardware) from Islamic Azad University, Iran, in 2008. His main research interests concern Artificial Intelligence, Question Answering Systems and Natural Language Processing.

Mojtaba khayat is a lecturer at the Shoushtar University; He received his M.Sc. degree in Computer Engineering (Computer Architecture) from Islamic Azad University, Iran, in 2010. He received his B.Sc. in Computer Engineering (Hardware) from Islamic Azad University, Iran, in 2007. His main research interests concern Artificial Intelligence, Multicast routing in wireless mesh network.