ANALELE UNIVERSITĂȚII
"EFTIMIE MURGU" REȘIȚA
ANUL XXI, NR. 1, 2014, ISSN 1453 - 7397

Sasan Saqaeeyan, Mohsen Shakibafakhr, Mohsen Roshanzadeh

Proposed an Optimal Search Algorithm to Find the Best Answer in a Question Answering Systems

QA systems extract answers in natural language question from a large set of documents. In this paper, we will design and implement Restricted Domain QA System based on a knowledge database. In this system we will use a genetic algorithm and optimal-genetic algorithm to search in the knowledge base for finding the answers. Web pages are sources of knowledge system. To validate the proposed approach, we will implement these algorithms; results indicate a significant increase in accuracy of the proposed system compare to previous systems.

Keywords: *Algorithms Optimal Genetic Algorithms, Question Answering Systems, Mutation, Crossover*

1. Introduction

Information Retrieval (IR) systems recover all documents that are related to the user query topic, by getting a few key-words from the user in a limited time. Documents that are retrieved by search engines from this way are related to the user's search in lexical aspect. This engine receives the user query that can be written in several form of keywords, and return documents which are relevant to user‵s questions. But it has some major problems: Firstly, users want to ask a question, but instead of their questions they should enter some keywords [1, 2].In this way, users have to usually convert their questions to the appropriate keywords, so users have some problems in converting and should learn some skills that it is a time-consuming work. In addition, only using a few key words cannot express clearly the purpose of users and converting will be impossible sometimes. So, using of keywords is not a perfect way to communicate between the system and users. On the other hand, users usually look for answers that are clear, while the output of the system is included a lot of documents that may not be an answer. Thus, users have to read a lot of documents. As a result of this, Information Retrieval System cannot fulfill centralized and precise information demands to prevent waste of users' time to read retrieved documents [3]. Thus a new research branch was formed by IR group which is called Questions and Answering (QA). In this system, a natural language question is given as an input to the system. The task of system is to find a precise short and complete answer for the questions in the shortest possible time. For this purpose, QA systems use both techniques of information retrieval (IR) and natural language processing (NLP) together [4].

QA systems are divided into two categories [5]: Limited domain which will answer questions in a specific domain and uses specific knowledge in that domain to process NLP. Open domain that deals with almost any type of question and can rely on global knowledge and public ontology.

Another classification that was proposed for QA was based on the number of language accepted by the system. A group of system which is called monolingual system receives a question of only one language and responds to it.

The rest of the paper is organized as follows. Section 2 presents an overview of QA systems and related works in this area. Section 3 is dedicated to the proposed work. Section 4 evaluates performance of the proposed system and compares it with the genetic algorithm. Finally, Section 5 exposes our conclusions.

2. Related Works

QA systems have three main components [6]:

1) Receiving and processing user queries and convert questions that are expressed in natural language to queries for using retrieval information component.

2) Retrieval information: search set of documents base on query from before step and retrieve documents.

3) Extract the final answer from retrieved documents.

All QA systems, have these steps, however, they use different methods to implement the process. The first devices to access information were retrieval systems for textual information. Still are very useful despite simplicity. Examples of these systems are Google, AltaVista and MSN Search that can be used to find intended documents in the internet. Number of information retrieval systems, are designed for use in textual sets in the web such SMART [7] and PRISE [8] systems. Web Question Answering System is another sample of QA system which uses genetic algorithms for ranking. In this system, by sending word to the Web, phrases are retrieved that are included answer. Set of retrieved sentences are matched with known previous paragraphs to extract new answer. Therefore, matching is an important parameter. Two strategies based on genetic algorithm, is proposed to improve matching.

(a) GASCA, is trained by syntactic patterns that concluded from pairs of (sentence, Answering). To match, and finding alignment, blocks of words is translated as adjacent blocks of zero and one that are adjacent. Then, according to their fitness, new blocks are made by function of "mutation" and "crossover" and make more possibility of matching a query with training patterns.

(b) PreGA, uses semantic communication, to match query and training pattern. Considering that, the previous strategy is based on syntactic patterns, if there is no sufficient syntactic pattern for suitable matching and textual pattern, answer is not shown clearly. However, using this strategy, query can be matched better with training pattern. [9].Basic algorithm has been used in the system which has been shown in Figure 1 [9].

```
Algorithm GA_QA
input: num_iter, pop_size, N, Q
begin
Rnd[1] ← 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 ←  Crossover (rnd[i],pc);
MAC ← Mutate (rnd[i], pm);
Rnd[i+1]←select Population(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. Algorithm. GA_QA

In this paper we investigate the GS_QA that is kind of QA system. This system can answer questions that are expressed in inventors' area. Knowledge base system is made from text and web pages. Then, to choose the best sentence in the knowledge base as an answer we use an optimal genetic algorithm. Accuracy of answer to questions is one of the advantages of the proposed system compare to QA systems that use genetic algorithms to search the knowledge base.

3. Architecture of GS_QA System

Structure and function of GS_QA, will be discussed in this section. this system has been implemented as a limited domain QA system. Like many similar systems, this system is composed of three main components. Main component of the system are: "Sentence processor system", "system of recovery and extraction answer "and "Ranking System". The general structure of system is shown in Figure 2.

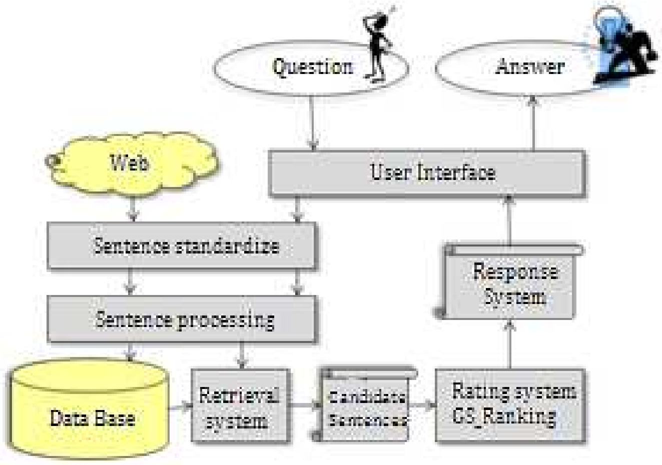

Figure 2. GS_QA system

Ranking System (GS_Ranking) is composed of several components. Figure 3, shows this system. In this section, by using of optimal genetic algorithm, candidate sentences are ranked and the sentence which has the highest rank is sent to output as a response.

3.1. User Interface

In this section, user questions in natural language, Num_iter and pop_size are received from input. Iter, is number of mutations. Bigger Iter number results to bigger mutations number and consequently finding an answer, is more likely. Size indicates initial population size that is selected for mutations with regard to the selective population in categories of four. So this number must be multiplied by four.

3.2. Standard of sentence

In this section, standardization is done on user questions that were entered in previous step. Standardization action is divided into three steps. First step: check all question words. Capital letters are converted to lowercase letters. Second step: all the extra words of {am, and, or, if, is, a, as, an, to, for, the} will be deleted from question sentence. Third step: to find the words which are ended {'s, es, er, ional, ion, ors, ive, ions, ed, or, ing}; all the words in the question sentence are investigated, and this prepositions will be deleted from terms and prepositions are removed from end of words.

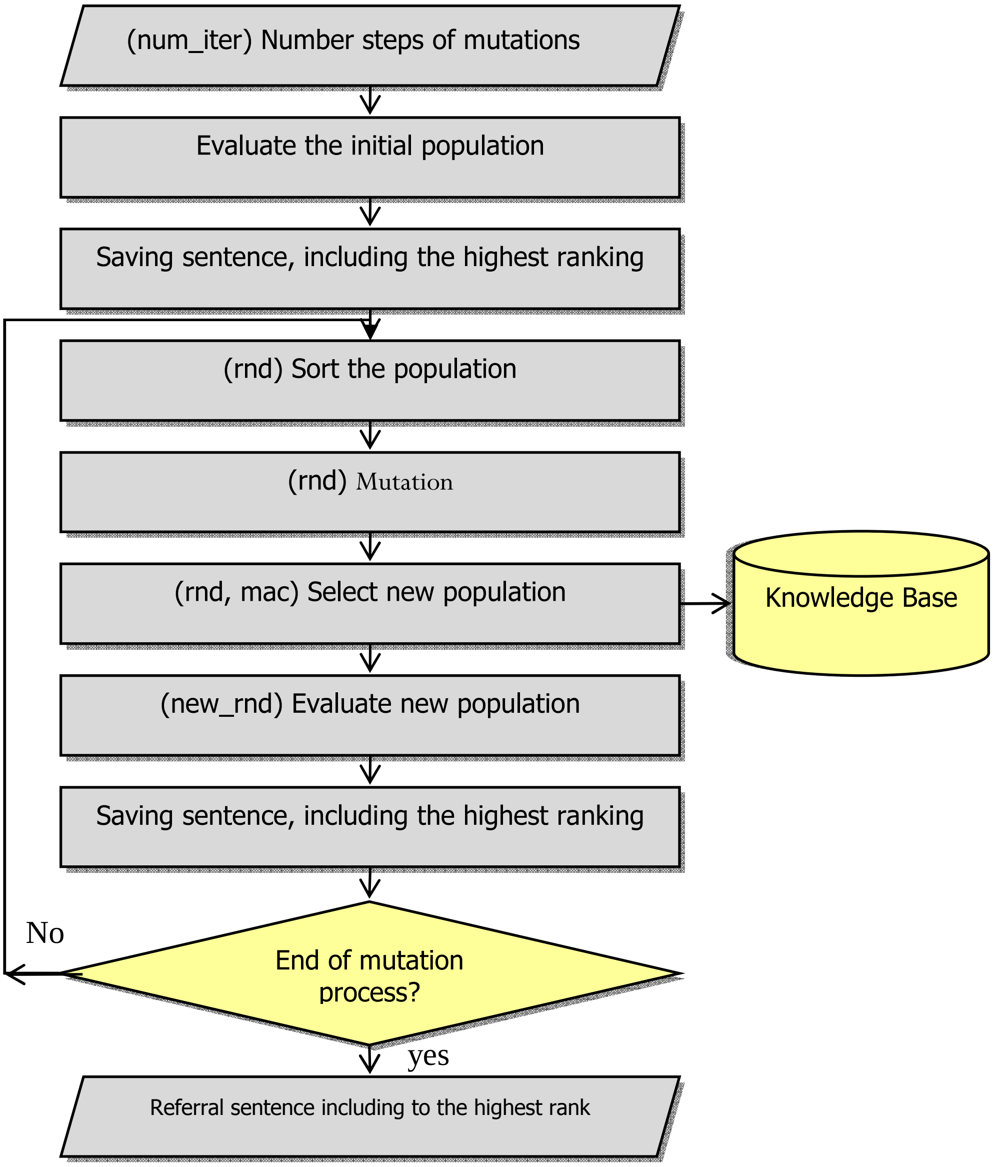
(num_iter) Number steps of mutations
Evaluate the initial population
Saving sentence, including the highest ranking
(rnd) Sort the population
(rnd) Mutation
(rnd, mac) Select new population
Knowledge Base
(new_rnd) Evaluate new population
Saving sentence, including the highest ranking
No
End of mutation process?
yes
Referral sentence including to the highest rank

Figure 3. GS_Ranking rating system

3.3. Sentence processor

In this section, the output from previous step is processed that contains all the keywords query. Type of question to be detected, and thus, regarding to the type of question answer can be guessed.

3.4. Retrieval System

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

3.5. Ranking System

After choosing selected phrases from the knowledge base in the previous section, those are being ranked. Candidate sentences are ranked base on matching keywords of query by selected sentences and matching type of question and type of selected sentence. The system is composed of five steps. Initial evaluation, sorting, mutations and crossover, second evaluation and referral answer.

Initial evaluation, here for the fitness of sentences of knowledge base are used two numbers (fit1, fit2). Fit1 is result of matching words of query, with all the words in existing sentences in the knowledge base. And fit2 is to investigate matching the question type with type of sentence of the knowledge base. These two values are calculated for all selected sentences. Global fitness, for sentence i, is calculated in the knowledge base, by equation (1).

$$\textit{Global fitness (i)} = (\textit{Fit1 (i)} * \textit{W1}) + (\textit{Fit2 (i)} * \textit{W2}) \quad (1)$$

W1 is coefficient of fit1 and its value is equal to 0.4, and W2 is coefficient of fit2, and its value is 0.6.Sorting in this step involves sorting arrays of Global fitness, and initial population, in an ascending order that is done by use of bubble sort. The crossover and mutations are done by optimal Genetic Algorithm. If a sentence has fitness more than 1.1 thus there is one phase for mutation and if fitness is between 0.4 and 1.0 there are four phases for mutation. And if fitness is between 0 and 0.3 mutation has 6 phases. Here the "reverse mutation" and "random mutation" are used. In Figure 4, GS_QA proposes an algorithm, which is used to improve the initial population

Algorithm 1: GS_QA

```
input: NIT, P, Q
    begin
      t = 1;  BestFound = 0 ;;
      pop[1] = create Initial Population(1, 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
```

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

Figure 4. GS_QA Algorithm

The second evaluation is done on new population that was generated in the previous step. Then all stages of the sort are repeated as many times as the num_iter. For reference answers, after the population being mutated as many times as the num_iter, a sentence that has the highest value of fitness is chosen as output and referrers to ranking system.

3.6. Display the final answer

This section is used to prevent displaying incorrect answers. Taking into account that the sentences which do not have a true answer also could have fitness equal 1, it prevents to displaying them in the output. So, if the highest fitness is less than 1.1, <NOT FIND> message is printed as the output. Otherwise, the sentence that has highest fitness is displayed as the output in the user interface.

4. Results of implementation

This section reviews the results of several experiments which have been done on the proposed approach by various data. Furthermore, proposed optimum genetic algorithm is compared with initial genetic algorithm. At first, user query and the size of initial population and mutation number and crossover operations, are entered then the candidate sentences of answer and eventually sentence which has the highest score is shown to user as an answer. In this paper, scoring methods of genetic algorithm and optimal genetic algorithm are compared with each other. Figure 5 shows fitness of implementation GS_QAsystem, with an initial population 4 and number of run of GS_QA is considered 10.

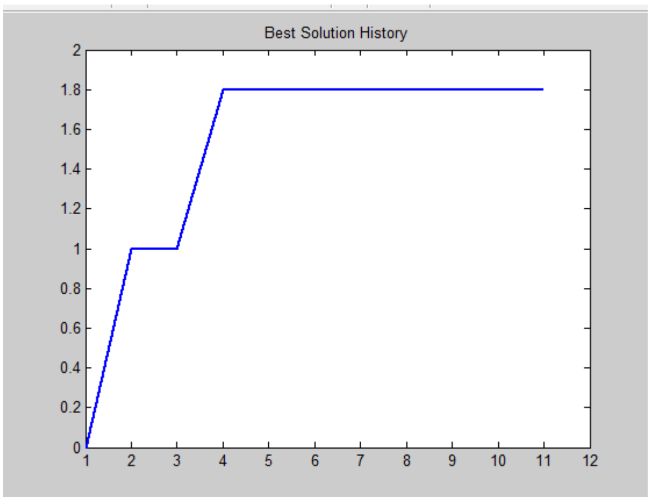

Figure 5 fitness of GS_QA System

As is indicated in the Figure 5, in the first step, fitness of the system is equal to 0, but fitness of system after several mutation and crossover get to 1.4, and by considering that this number is greater than 1.1, so the system has found its answer. Figure 6 show fitness of GA_QA system and GS_QA, with an initial population 12 and the number of implementation 1 to 10.

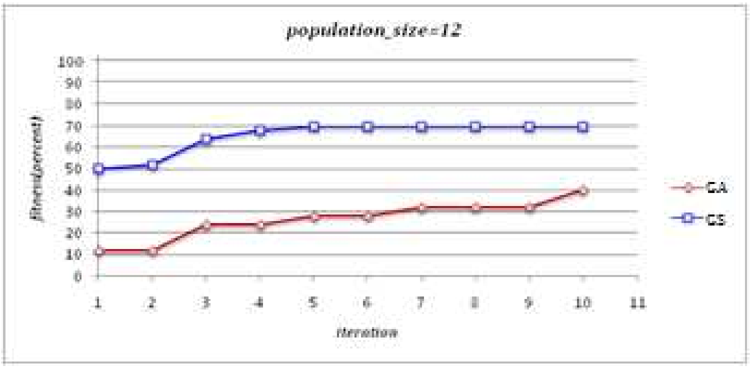

Figure 6. Comparison of fitness GA_QA system and GS_QA, with an initial population 12

In Figure 7 in the fifth step, average fitness of GS_QA system, and so the accuracy of the system is 70%, which has improved 42 percent compare to the fitness of the system GA_QA, (28 percent).in Figure (7) average percentage of GA_QA fitness and GS_QA fitness, with number of implementation from 1 to 10, is shown.

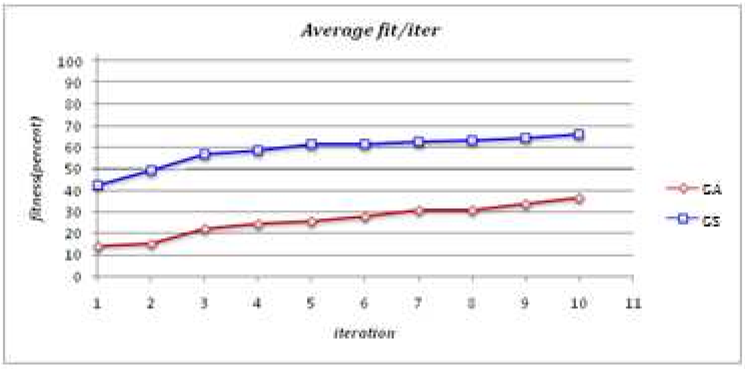

Figure 7. Compare between average percentage of GA_QA system fitness and GS_QA fitness, based number of implementation actions from optimal genetic algorithm

5. Conclusion

In this paper, a restricted domain question of answering system, which was developed based on the knowledge base, was presented. This system uses the optimal genetic algorithms for ranking. Knowledge base system has composed of structured texts to build knowledge base that uses unstructured web pages. Standardized components and the sentence processor of system convert natural language query to keywords for scoring sentences in the knowledge base by using these keywords. Scoring method is based on matching of query keywords with sentences in the knowledge base, as well as type of questions in matching with type of knowledge base sentences. After scoring to sentences in the knowledge base, sentence which has the highest rate is displayed via user interface in output. In this paper, Performance of the system was investigated by implementing of optimal genetic algorithm and genetic algorithms. According to evaluations, the average of accuracy of the proposed system compare to GA_QA has improved significantly.

ACKNOWLEDGMENT

This paper derived from research project "Proposing an Optimum Search Algorithm to Find the Best Answer in a Question Answering System". Done in Abadan Branch, Islamic Azad University, Abadan, Iran.)

References

[1] Oh H., Sung K., Jang M., Myaeng S., *Compositional question answering: A divide and conquer approach*, Information Processing & Management, Volume 47, Issue 6, November 2011, Pages 808–824.

[2] Moussa A.M., Abdel-Kader R.F., *QASYO: A Question Answering System for YAGO Ontology*, International Journal of Database Theory and Application Vol. 4, No. 2, June 2011, pp. 99-112.

[3] Li X., Roth D., *Learning question classifiers*, The 19th International Conference on Computational Linguistics, 2002, pp. 556–562.

[4] Ghobadi-Tapeh A., Rahgozar M., *A knowledge-based question answering system for B2C ecommerce*, Elsevier, Knowledge-Based Systems 21 (2008) 946–950.

[5] Baayen H., Planck M., Klavans J., Barnard D., Tufis T., Llisterri J., Johansson S., Mariani J., *Advances in Open Domain Question Answering*, Text, Speech and Language Technology, Volume 32, Netherlands, 2008.

[6] Téllez-Valero A., Montes-y-Gómez M., Villaseñor-Pineda L., Peñas Padilla A., *Learning to select the correct answer in multi-stream question answering*, Springer, Information Processing and Management, Volume 47, Issue 6, November 2011, Pages 856–869.

[7] Salton G., *the SMART Information Retrieval System*, Prentice Hall, Englewood Cliffs, NJ, 1971.

[8] Dimmick D., O'Brien G., Over P., Rogers W., *Guide to Z39.50/Prise 2.0: Its Installation, Use & Modification*, Gaithersburg, Maryland, USA, 1998.

[9] Figueroa A.G., Neumann G., *Genetic Algorithms For Data-Driven Web Question Answering*, Evolutionary Computation, Volume 16 Issue 1, MIT Press Cambridge, 2008.

[10] Oh, H.-J., Sung K, Jang M, Myaeng S, *Compositional question answering: A divide and conquer approach*, Elsevier, Information Processing and Management, Volume 47, Issue 6, November 2011, Pages 808–824.

Addresses:

- Sasan Saqaeeyan, Department of Computer, Abadan Branch, Islamic Azad University, Abadan, Iran, sasan_sagha@yahoo.com
- Mohsen Shakibafakhr, Sama Technical and Vocational Training College, Islamic Azad University, Shoushtar Branch, Shoushtar, Iran
- Mohsen Roshanzadeh, Department of Computer, Abadan Branch, Islamic Azad University, Abadan, Iran