A Natural Language Question Answering System- an Application for e-learning

Akash Gupta[a] and Prof. V. Rajaraman[b]

[a]*Synopsys India Pvt. Ltd.*
131/82/2C, 3[rd] A Cross, 18[th] Main, Koramangala, Bangalore 560095, India
Tel: +91-80-5522201, E-mail: akgupta@synopsys.com

[b]*Professor, Jawaharlal Nehru Centre for Advanced Scientific Research*
Bangalore 560012, India

Abstract

This paper describes a natural language question answering system that can be used by students in getting a solution to their queries. Unlike AI question answering systems that focus on the generation of new answers, the present system retrieves existing ones from question-answer files. Unlike information retrieval approaches that rely on a purely lexical metric of similarity between query and document, it uses a semantic knowledge base (WordNet) to improve its ability to match question.

Paper describes the design and the current implementation of the system as an intelligent tutoring system. Main drawback of the existing tutoring systems is that the computer poses a question to the students and guides them in reaching the solution to the problem. In the present approach, a student asks any question related to the topic and gets a suitable reply. Based on his query, he can either get a direct answer to his question or a set of questions (to a maximum of 3 or 4) which bear the greatest resemblance to the user input. We further analyze application fields for such kind of a system and discuss the scope for future research in this area.

Keywords – Intelligent tutoring system, knowledge based question- answering, Natural Language Processing.

1. Introduction

Natural language processing is an important sub-area of research in the field of Artificial Intelligence. The problem is to develop an effective natural language front-end for a database system. That is, there is a program, which translates the query the user has given from natural language to some database query language – an artificial language designed to be used by computer databases.

The ability to communicate in some kind of natural language, be it English or French, often seems to be the hallmark of the human race. Understanding natural language requires both linguistic knowledge of the particular language being used and world knowledge relating to the topic being discussed. When a teacher delivers a lecture on any subject, it is usually followed by some time allocated for question answering. Any problems that a student might have encountered or any part of the lecture, which he failed to comprehend, can be explained by the teacher during this interval. But in an entirely different scenario when a student tries to grasp the same topic from a book, questions which come up in his mind (more so if he is new to the topic) usually remain unanswered. It is this problem that I intend to tackle in this paper.

The World Wide Web offers an unprecedented opportunity for producers and consumers of intelligent tutoring systems. Many authors now prefer to provide their books through the net as electronic books (e-books). The major problem with this approach is that since there is no feedback mechanism involved, it often results in the student cramming up the whole thing rather than understanding. There is a need for a question answering mechanism where a student can place his doubts, queries related to the topic and get instant answer.

Database used for this purpose contains questions pertaining to a specific subject. The question serves as an index to the knowledge contained in the answer. The user gains access to the information stored in the database by querying it in English. The system analyzes the query and decides through a matching process what question in the database is similar in context to the question posed by the user. Then it retrieves the corresponding answer to the question from the database.

2. Related work

Many models for natural language question answering systems has been proposed before, for example START by Katz [1] and FAQ finder by Burte *et al.* [5]. FAQ Finder returns a set of 5 questions bearing the greatest resemblance to the user query. This approach even though increases the probability of getting a correct answer is ill suited for application in the tutoring field. In contrast with this, our present approach aims at increasing the probability of providing a direct answer on as many occasions as possible.

In the FAQ Finder system, each Q&A pair is turned into a vector of terms whose weights are standard tfidf values (term frequency times log of inverse document frequency) but if a particular keyword occurs in many Q&A pairs of a FAQ (frequently asked questions file), it's tfidf value would be insignificant thereby resulting in the matcher not retrieving that Q&A pair.

Further, whenever new questions have to be added to a FAQ file, idf value (inverse document frequency) of all possible terms in every Q&A pair in that file would have to be recalculated. This introduces large processing overhead for offline indexing.

Boris Katz [1] in his natural language system START (SynTactic Analysis using Reversible Transformations) has described how indexing and retrieval of information can be done using subject-relation-object triples decorated with what he calls a history. The understanding module of START analyzes text and indexes the knowledge contained within it into a knowledge base. It uses it to generate answers to users query. One of the applications of a tutoring system is to supplement the information provided by the author in his/her book. Thus response to a user's query might have to be more explanatory and not solely limited to the content of the book. To perform the same task using START might involve feeding a large repository of information related to the subject matter into the database, which might be called a book in itself.

Further, START combines subject, object and the relation between them into ternary expressions (T-expressions) which "summarizes' the syntactic structure of the sentence. Thus it is more suited for generating single line responses. Since in a tutoring system, most of the answers are fairly long, START module is not adequate for the present approach.

Aim of most of the tutoring systems available at the moment like ADIS (Animated Data Structure Intelligent Tutoring System) described by Warendorf et al. [6] or Andes by Kurt VanLehn et al. [7] is to help students in understanding methods of solving certain predetermined problems. No attempt is usually made to provide the students with an opportunity to put across their own queries rather such tutoring systems can be compared to an interactive exercise book that offers a practice environment with the benefit of a coach providing advice and remediation and keeping students on track toward solution to a particular problem.

A major advantage of our model is its conceptual simplicity. It also enhances the flexibility and extensibility of our core database without too much increase in the offline processing time of the computer.

3. Problem Specification

The idea behind such kind of a database is to record a suitable answer to some common question and make that answer available, particularly to newcomers who may otherwise ask the same questions again and again. The question serves as an index to the knowledge contained in the answer. If the question happens to be similar to one of the frequently asked ones whose answer has been recorded in the database, the present system is likely to return the appropriate answer. Thus in short, the problem that we are considering here can be stated as follows.

Given a question pertaining to the subject matter, our software must be able to analyze it, break it up into its various components, extract information from those components based on the existence of some predefined keywords (words that are specific to a topic) plus a binding relation and bring out the relevant answer from the database pertaining to the question asked. In case the question is not very specific, it should bring out all the questions from the database which are related in someway or the other to the user query. Thus system must be able to handle all kinds of natural language queries (including those which are not grammatically correct but contain sufficient information to convey the kind of question user has in mind), which are posed by the user.

4. Outline of approach

Most English sentences break up into three units- subject, object and a connecting relation. Subject and object of a sentence are self- explanatory. Relation may be a verb, adjective or even a noun linking the object of a sentence with its subject. Consider the following question:

How can plaintext be converted into ciphertext?

'plaintext' and 'ciphertext' are the subject and object respectively while 'convert' is a relation joining the two. Notice that here relation has been taken as the basic form of the actual word (i.e. 'converted') in the sentence. Further there has to be some kind of differentiation between a yes-no question, and wh-question like whom-questions, where-questions, why-questions, etc. As an example consider the following two examples:

1) How can we obtain ciphertext from plaintext?

2) Can we obtain ciphertext from plaintext?

Both of them have the same values for the subject, object and relation fields but their answers are different. In the first case, we have to describe the steps involved in converting a plaintext into a ciphertext while the second one merely requires a yes-no kind of answer. So, to handle this I have introduced a fourth field into the database which contains the word from which the question starts like who, what, can etc.

5. Design of the database

The present module has been developed for a still to be published book by Prof. V. Rajaraman, Honorary Professor, SERC, IISc Bangalore. Pertaining to each chapter in the book, there is a separate Expert system incorporates the question answering section developed as part of the present approach.

The database has been divided into a set of tables. One of them contains all possible questions pertaining to a specific topic along with their solutions. There are three fields namely pointer value, question and answer. Associated with each question is a pointer field that is used for mapping purpose from other tables. The other table in the database consists of a field containing the first word that the user has entered (referred as keyword 1), subject (Keyword 2), object (Keyword 3) and a relation field. I have tried to put the possible relations for each question into the relation field. All of these different entries pertaining to the same pointer field value will map to the same question in the question- answer table. Now continuing with our previous example, the same question can be asked in a multitude of ways:

1) How can we obtain ciphertext from plaintext?

2) How can ciphertext be obtained from plaintext?

3) How can plaintext be converted into ciphertext?

4) How can we convert plaintext into ciphertext?

Here if I put 'obtain' in the database then the expert system can recognize 'obtained' and 'obtaining' as the different tense forms of the word 'obtain' by carrying out morphological analysis.

Besides morphological analysis, there is one more classification based on the different Domain-specific words like IP address, Web address or Internet address that convey the same meaning. This also has to be kept in mind while preparing the database. To handle this, 'IP', 'Web' and 'Internet' will all go into Keyword 2 field whereas Keyword 3 will consist of 'address' for all the three record entries sharing the same pointer value.

Another classification of words that can come handy in this case is the knowledge of synonyms. For this, we require the help of a semantic database (WordNet) to improve the program's ability to match question posed by the user with the one in the database. WordNet is capable of providing us with all possible synonyms (by virtue of its large database consisting of 1,73,839 different words). Still I would have to put atleast those relations into the relation field, which are neither related to each other as morphological variants, and neither as synonyms or words with somewhat similar meaning. Thus for the handling 1-4 questions given above, I would have to put both 'obtain' and 'convert' into the relation field thereby leading to a many to one mapping from one table in our database to another.

6. Use of Semantic Knowledge Base-WordNet

The WordNet System consists of lexicographer files which organize nouns, verbs, adjectives and adverbs into groups of synonyms, and describe relations between synonym groups. These relations include hypernymy, hyponymy, antonymy, entailment, meronymy and holonymy. Information in WordNet is organized around logical groupings called synsets such that they are interchangeable in the same context.

All that the expert system needs and requires are words that are related to each other via the synonym or similar links. So WordNet has been used to build another database with the Tables in it pertaining to synsets. It essentially consists of one central Table named 'Dictionary' which is an assimilation of all words or collocations (example "fountain pen", "take in") in the database along with their pointers. Then there are separate Tables pertaining to verb synsets, adjective synsets, etc. that convey similar sense and meaning. These Tables simply contain the corresponding pointer values.

7. Implementation details

Well, my approach has been to try to provide a direct answer to the user query in as many instances as possible. As stated earlier, Keyword 2 and keyword 3 entries are a prior requisite for conducting a search and in case both of them are absent, the user will be prompted to reframe the question inline with the content of the section he is presently dealing with.

In short, the program will first tokenize the sentence (i.e. break up into the different words). No processing is needed when there is an exact string match between a user's question and a question in the database in which case answer is presented directly. Otherwise, we begin by conducting a search for the Keyword 2 entry. stored in an array say 'A'. A search is now made for the entries from Keyword 3 having a corresponding Keyword 2 entry. We store them in an array 'B'.

If no Keyword 2 entry is found, we directly move onto searching for the Keyword 3 entries putting them in array 'B'. After this, the program searches for the 'valid' relation field entries having the corresponding Keyword 3 entries in array 'B'.

Given a query, the program will carry out the following sequence of steps for generating a direct answer:

1) First, it will test whether or not a single entry from Keyword 2 and keyword 3 has been entered since they are domain specific and having more than one entry can alter the meaning completely. The problem arises if for a particular Keyword 2 value, there are multiple distinct Keyword 3 entries one of which is 'null'. For example consider the case when the question is:

What is e-commerce?

Entries in Keyword 3 field are 'null' and 'mean'. If a person enters the question
What do you mean by ecommerce?

In this case, there will be 2 valid entries in array 'B'. In general, if array 'B' contains any non-null or if for a particular Keyword 2 entry there can only be a 'null' value in Keyword 3 field, we will proceed to step 2.

2) After this, we make comparisons for the relation field entry. The entry should have the same pointer field value as an entry in array 'B'. If array 'B' is empty or if array 'B' has got only 'null' entry then it is presumed that no valid keyword 3 entry has been found. In such a case, we make use of entries in array 'A' for finding the relation and store them in an array 'C' (say).

We search for the morphological variants of the relations in the user input and put them in an array 'D'. If array 'C' and 'D' are empty, the system will get the synonyms for the valid relations from the database. Valid relations are those relation field entries for which we have an entry either in array 'B' or if no Keyword 3 is found then in array 'A'. These synonyms (stored in a separate array 'E') will then be compared against the unrecognized words for a match. Condition for getting a single relation is that there should be only a single entry either in array 'C','D' or 'E'. If both array 'C' and 'D' are empty but 'null' is allowed as a valid relation then also 'null' would be considered as a single relation and we will move onto step 3.

3) After this, it checks the first word of the user query to ascertain whether it is a valid question or merely a statement. If it matches the corresponding entry of the Keyword 1 field for which we have found a single relation, direct answer is provided to the user.

Certain precautions are necessary in the design of the database:

1) If for 1 record, we have set say 'payment' as a Keyword 3 entry and another record also has 'payment' as one of the Keywords, avoid placing it as a Keyword 2 entry. Since in that case if payment is detected in a user query along with it's corresponding Keyword 2 entry then count 2 would atleast be two and so in accordance with the above 4 criterion direct answer would not be possible.

2) Try to generate all possible Keyword 2 and Keyword 3 entries pertaining to a particular question since we are not carrying out any morphological analysis or synonym checking for them. Thus for the particular question:

What is required for Electronic Data Interchange?

Not only there is a need for placing Electronic data interchange as a Keyword 2 or Keyword 3 entry but also an entry for EDI (which is it's acronym) has to be there.

3) Auxiliary verbs, prepositions such as by, in, to, from, with etc., pronouns such as I, thou or you, he, she, it, we, ye, and they should not be included as relation field entries. Thus if no proper relation is available, then put 'null' in the relation field Such questions when posed by the user would generate direct answer even if they lack a relation. At the same time, one should also try to avoid overuse of 'null' entry since it might reduce the hit probability of getting a direct answer.

4) Relation field must contain only the root form of the word This is essential for successfully making the morphological analysis. We would therefore have 'obtain' in place of obtained, obtains or obtaining and 'withdraw' rather than withdrawn, withdrawing etc.

8. Generating a set of questions

Aim of this section is to reduce the number of choices presented to the user to a maximum of 3 or 4. Various combinations of inputs given by the user need to be considered. Thus the choices generating mechanism can be divided into several parts:

1) Keyword 2 is found but Keyword 3 & relation are not found.

2) Keyword 2 and Keyword 3 are found but relation is not found.

3) Keyword 2 and relation is found but Keyword 3 is not found.

4) Keyword 3 and relation is found but Keyword 2 is not found.

5) Keyword 3 is found but Keyword 2 & relation are not found.

6) Multiple entries from all the 3 fields are found in user input.

Here, entries from Keyword1, Keyword 2, keyword 3 and relation field may not be corresponding to each other. The expert system must be able to handle all kind of user query including improperly framed or incomplete questions.

Case 1:

If there is no non-null relation in array 'C', also array 'D' and 'E' are empty then we take all the Keyword1 entries, which have the same pointer value as the entries in array 'A'. These will now be compared against the first word entered by the user, since often it holds the key to the kind of question being asked (for example whether it is a yes-no question or wh-question). If a match is found then only those questions would be displayed for whom the value in Keyword1 matches the first word of the question entered by the user else questions corresponding to all the entries in array 'A' would be displayed.

Case 2:

If corresponding Keyword 1 entry matches the first word of the user input then only those questions would be displayed else we have to display all the questions for which we have the corresponding Keyword 2 and Keyword 3 entries.

Case 3:

Let array 'F' consist of pointer field values corresponding to the entries in array 'C', array 'D' and array 'E'. We already have an array say 'pkey2' that contains the corresponding pointer values for the entries in array 'A'. We take the pointer field values common to both. This is necessary to eliminate the following anomaly:

When we gradually narrow down our question base by proceeding from Keyword2 then to Keyword3 and finally reaching upto relation field, we might land up with a relation that is also present in a record having different Keyword2 or Keyword3 values. So, pointers corresponding to those extra relations would also creep in and need to be eliminated.

For a corresponding entry in Keyword2, if any of theKeyword1entry matches the first word of the user question then we want to consider only pointer value corresponding to that.

Case 4:

As in the previous case, we will begin with an array which consists of all the entries from array 'C', array 'D' and array 'E'. We will run a loop for all the entries in array 'F'. We already have an array say 'pkey3' that contains the corresponding pointer values for the entries in array 'B'. Now we will consider only those pointer field values in array 'F' for which we have an entry in 'pkey3'. Also, if a corresponding entry in Keyword1 matches the first word of the user query we consider only the pointer value corresponding to that.

Case 5:

Run a loop considering entries found in Keyword3 field one at a time. For this case we will begin by first storing all the distinct pointer field values corresponding to the entries in array 'B'. If the pointer corresponding to the first word entered by the user happens to be among these, we neglect all the other pointer values for this particular Keyword3 entry. Otherwise we display the appropriate questions.

Case 6:

If any entry from Keyword 3 field is detected, search is made for the valid relation field entries on the basis of the entries in array 'B'. This can create a problem if user question contains relation field and Keyword 2 entries from one record and the Keyword 3 corresponding to another record .

To handle this case, if there is more than one entry in array 'A' and an entry from Keyword 3 field is also detected following analysis is carried out. For each entry in array 'A' if corresponding entries from Keyword 3 are present than those entries are used for finding the valid relation else we use that particular entry from Keyword 2 itself to find all logically possible relations depending on the pointer field value. Even now if no non-null relation value is found and array 'D' and 'E' are still empty, we can safely say that user question contain no valid relation field entry. Here also six more cases are possible:

1) Keyword 2 is there but corresponding entries from Keyword 3 and relation field are absent.

2) Keyword 2 and relation field values are present but corresponding Keyword 3 entry is absent

3) Keyword 2 and Keyword 3 values are present but corresponding relation field entry is absent.

4) All the three Keyword 2, Keyword 3 and relation field entries are present.

5) Keyword 3 and relation field values are present but corresponding Keyword 2 entry is absent.

6) Keyword 3 is there but corresponding entries from Keyword 2 and relation field are absent.

Now run a loop for pointers corresponding to all the Keyword 2 values entered by the user. For each pointer value if the user has entered corresponding Keyword 3 value then choose a variable let's say Key3found to 1. Also if corresponding relation field value (either direct relation or it's morphological variant or it's synonym) has been entered by the user then set another variable let's say Relationfound to 1. Now if Key3found and Relationfound are both zero then do similar analysis as was done for Case 1. Next if Key3found is 1 but Relationfound is zero then do the analysis to the analysis for Case 2 and if Key3found is zero while Relationfound is 1 then do the analysis similar to the analysis for the Case 3.

9. Analysis of the system

In this subsection we look at the performance of our system with respect to various possible kinds of user inputs. Each example is followed by a brief explanation of the result obtained.

1. How does electronic cheque payment work?

Ans: Cheque payments are made between organizations using digitally signed cheques and public key certificates issued by a certifying authority.

Analysis: Note that in the above case, uppercase and lowercase letters hold the same significance. Here entries in Keyword 2, Keyword 3 and relation field are 'cheque', 'payment' and 'work' respectively.

2. How can we find ciphertext from plaintext?

Ans: Ciphertext is the encrypted form of plaintext. It is obtained from plaintext by applying transposition & substitution using keys.

Analysis: In this case 'find' is a synonym of the word 'obtain' which is present in the database. This result shows that synonyms are also equally accepted by the system.

3. What is a proxy application gateway?

Ans: Sorry! I know only answer to this:

What is the advantage of proxy application gateway?

< Answer to this question is simultaneously displayed in the answer window >

Analysis: Keyword 2 and Keyword 3 entries in this case are 'proxy' and 'gateway'. No Entry for Keyword 3 field is obtained in the user input but since there was only one question containing the given Keywords, both the question and the answer were displayed to provide greater assistance to the user of the system.

For the remaining examples, entry in Ans: section would depict the choices of questions given by the system. When the user clicks anyone of them then the corresponding answer is displayed in the answer window.

4. What are e-cash and firewall?

Ans: How does electronic cash payment take place?

How does a firewall enhance the security of a LAN?

How do we transfer coins in electronic cash payment?

What is the use of hardened firewall host?

Analysis: This pertains to the Case 2 that we had discussed earlier on. 'e-cash' is an entry in Keyword 2 field while 'firewall' is an entry in Keyword 3 field in separate records. No relation field entry is obtained.

5. How can firewall enhance the security?

Ans: How does a firewall enhance the security of a LAN?

What is the use of hardened firewall host?

Analysis: This pertains to the Case 3 that we had discussed earlier on. Corresponding Keyword 2 entry ('firewall') and relation field entry ('security') are present but the Keyword 3 entry is missing.

6. How can security in a LAN be enhanced?

Ans: What is a LAN?

How does a firewall enhance the security of a LAN?

Analysis: This pertains to the Case 4 that we had discussed earlier on. Corresponding Keyword 3 entry ('LAN') and relation field entry ('security') are present but the Keyword 2 entry is missing.

7. What is an organization?

Ans: Sorry! I know only answer to this:

How does a firewall enhance the security of a LAN?

< Answer displayed in the answer window >

Analysis: This pertains to the Case 5 that we had discussed earlier on. Only a single entry from Keyword 3 field is present whereas there is no entry in the user question from Keyword 2 or the relation field.

10. Shortcomings of present approach

The effectiveness of the approach lies in the design of the database. If precautions highlighted earlier are not adhered to, system might provide put forth a set of choices in response to a user query even though a direct response might be plausible.

The expert system also suffers from its slow reaction. However, deleting some of the rules in the system can make the speed faster, but at the same time, the number of undiagnosed error would increase. On the other hand, more rules can be added to the system to increase the probability of a correct response. But the system will require more time to search for the right rule, which ill result in a slower response.

11. Conclusion

A Question - Answering system has been described that handles the query posed by the user and tries to match it with the information contained in the question stored in the Database. Based on this match, system either provides the user with a direct answer or displays a set of questions (3 or 4 maximum) that bear the greatest resemblance to the user query. Main aim of the approach employed for tackling the problem has been to try to reduce the number of choices given by the system and to the maximum extent possible try to provide the user with a direct answer. Our evaluation although conducted with a small corpus of questions, demonstrates the effectiveness of the system.

The power of our approach arises from the fact that we are using a repository of questions designed specifically to answer the commonly asked domain specific questions and as such are more highly organized than free text.

Thus for an application such as answering student queries pertaining to a particular chapter (Domain specific) this approach seems to be viable. Such a system can also be used on company websites to answer customer queries pertaining to a particular product. This can increase the interactivity of the site and help it in gaining greater recognition. On the same hand, it would also serve to reduce the burden of the public relations cell in the company as the more commonly asked customer questions could directly be handled by the proposed system.

Using a larger database we can further enhance the effectiveness of the system. Greater are the number of questions in the database, greater will be the number of the direct responses the system would be capable of generating. But for each added question in the database, we would have to create the corresponding entries (i.e. the entries for Keyword 1, Keyword 2, Keyword 3 and relation field). Thus as the number of entries increases linearly, database of possible keyword values tend to increase at a rate approximately twice of that.

An easier way of handling the situations is to form certain set of rules according to which entries for the fields (i.e. Keyword 1, Keyword 2, Keyword 3 and relation field) can be generated at run time. To illustrate this fact, let us take an example:

What is e-commerce?

What is a Digital signature?

In both the cases, Keyword 1 and relation field entries would remain the same only change that is occurring is in Keyword 2 and Keyword 3 field. So under the rule <Definition of a term> we can have only the Keyword 2 and Keyword 3 entries different for different questions let's say pertaining to definitions of terms. By formulating such kind of rules for all possible questions we can greatly reduce the size of the database. Thus the future research on this system can be carried out along that direction.

12. References

[1] B. Katz (1988). Using English for Indexing and Retrieving. In P.H.Winston and S.A.Shellard (Eds.), *Artificial Intelligence at MIT Expanding Frontiers* (vol. I, pp. 134-165). MIT Press, Cambridge, MA.

[2] E. Charniak, D. McDermott (1998). *Introduction to Artificial Intelligence.* Addison Wesley Publishing Company, India.

[3] Terry Winograd (1983) *Language as a Cognitive Process* Vol. I Syntax, Addison Wesley Publishing Company, Reading, Massachusetts.

[4] Beckwith, Richard and George A. Miller (1990) Implementing a lexical network. *International Journal of Lexicography,* 3 (4), 302 - 312.

[5] Burke, D. Robin, Kristian J. Hammond, Vladimir Kulyukin, Steven L. Lytinen, Noriko Tomuro and Scott Schoenberg (1997) Question answering from frequently asked question files --- experiences with the FAQ FINDER system. *AI Magazine,* 18 (2), 57 - 66.

[6] Warendorf, K. & Tan, C. (1997), ADIS-An animated data structure intelligent tutoring system, or Putting an interactive tutor on the World Wide Web, In *Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web,"* 8th World Conference of the AIED Society.

[7] Schulze, K.G., Shelby, R.N., Treacy, D.J., Wintersgill, M.C. (2000). Andes: A Coached Learning Environment for Classical Newtonian Physics. In *Proceedings of the 11th International Conference on College Teaching and Learning.* Jacksonville, FL.