Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 30 (2012) 1187 – 1194

Procedia Engineering

www.elsevier.com/locate/procedia

International Conference on Communication Technology and System Design 2011

An Approach for Extracting Exact Answers to Question Answering (QA) System for English Sentences

Raju Barskar[1], Gulfishan Firdose Ahmed[2], Nepal Barskar[3], 1*

[1]CSE Department ,University of Institute Technology,
Rajiv Gandhi Prodhyogiki Vishvidhyalaya, Bhopal (M.P.) Pin-462036, India.
[2]CSE&IT Department
Maulana Azad National Institute of Technology, Bhopal (M.P.) Pin-462051, India.
[3]CSE Department, University of Institute Technology,
Rajiv Gandhi Prodhyogiki Vishvidhyalaya, Bhopal (M.P.) Pin-462036, India.

Abstract

Recent advances in Natural Language Processing (NLP) and AI are trying to build systems to the point where people may converse with a machine in natural language to get answers to their questions. The question answering system based on keywords search. This is similar to Web search. We are developing a Question Answering (QA) System for English sentences. The user should be able to access answer of their questions in a user friendly way, that is by questioning the system from the given English paragraph and the system will return the intended answer by searching in context of the paragraph using the repository of English dictionary.
In this paper we present a Question/Answering system that takes advantage from category information by exploiting several models of question and answer categorization.A novel strategy, in addition to conventional search and NLP techniques, will of be used to construct the QA system. The focus is on context based retrieval of information. This paper provides a novel and efficient method for extracting exact textual answers from the returned documents that are retrieved by traditional IR system in large-scale collection of texts. For testing purpose, the proposed methodology is applied in text classification and the accompanying experimental results are compared with the output provided by a probabilistic based approach.

Keywords: - NLP, QA System; Predicate Logics, Knowledge Representation; Clause form, Herbrand's Theorem; Pattern Extracting;

Introduction

Natural Language Processing (NLP) is the computerized approach to analyzing text that is based on both a set of theories and a set of technologies. And, being a very active area of research and development, there is not a single agreed-upon definition that would satisfy everyone, but there are some aspects, which would be part of any knowledgeable person’s definition. The definition is Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. The goal of NLP as stated above is “to accomplish human-like language processing”. The choice of the word ‘processing’ is very deliberate, and should not be replaced with ‘understanding’. For although the field of NLP was originally referred to as Natural Language Understanding (NLU) in the early days of AI, it is well agreed today that while the goal of NLP is true NLU, that goal has not yet been accomplished. A full NLU System would be able to:

1. Paraphrase an input text
2. Translate the text into another language
3. Answer questions about the contents of the text

* Raju Barskar. Tel.: +91-9893181206
E-mail address: rajubarskar7863@gmail.com.

doi:10.1016/j.proeng.2012.01.979

4. Draw inferences from the text

NLP techniques are used in applications that make queries to databases, extract information from text, retrieve relevant documents from a collection, translate from one language to another, generate text responses, or recognize spoken words converting them into text. From the QA point of view, NL interfaces to databases [3] and information extraction [7] are most interesting. A common feature of NLP systems is that they convert text input into formal representation of meaning such as logic (first order predicate calculus), semantic networks, conceptual dependency diagrams, or frame-based representations [2]. Since the early days of NLP, (QASs) systems simulated human intelligence within the NL understanding research field. They worked as NL front-end to databases [8], dialogue systems [11] or story comprehension systems [2-3].*

Question Answering System

Research in Question-Answering (QA) is not new. The QA problem has been addressed in the literature since the beginning of computing machines. The AI/NLP communities initiated traditional work to address question-answering using structural methods. Early experiments in this direction implemented systems that operate in very restricted domains (e.g. SHRDLU [Winogard, 1972] and LUNAR [Woods, 1972]). In the QUALM system, Lehnert [1978] took a further step, based on the conceptual theories of Schank & Abelson [1977], to understand the nature of the questions and classify them in a way similar to how human beings understand and answer questions. SCISOR [Jacobs & Rau 1990] aimed at question answering and text extraction more than information retrieval. It combined natural language processing, knowledge representation, and information retrieval techniques with lexical analysis and word-based text searches. The MURAX system [Kupiec, 1993] used robust linguistic methods to answer closed-class natural language questions.

It presented the user with relevant text in which noun phrases are marked. A less automated approach like Ask Jeeves [1996] approached the QA problem by pointing the questioner to Web links that might contain information relevant to the answer to the question. Ask Jeeves benefited from advanced natural language processing techniques combined with data mining processing and a huge expanding knowledge base. Another system, with a different approach, is the FAQFinder system [Burke et al., 1997], which attempted to solve the question-answering problem using a database of question-answer pairs built from existing frequently asked question (FAQ) files. Two other important systems are the START system [Katz, 1997], which is based on annotations from the Web and the Q&A system [Budzik & Hammond, 1999], which is a semi automatic, natural language question answering and referral system. The system is based on a huge knowledge base and human experts who volunteered their time to respond to the users' questions.

Question answering (QA) is the task of automatically answering a question posed in natural language. The question answering system based on keywords search. This is similar to Web search. There are many disadvantages of Web search. It gives lot of web pages for searching a single word. It is time consuming. It takes more time for extracting relevant document. Several known "futurists" believe that computers will reach capabilities comparable to human reasoning and understanding of languages by 2020. Automated Question Answering (QA), the technology that locates, extracts, and represents a specific answer to a user question expressed in natural language, is now being studied very actively by researchers and practitioners. We believe that automated QA technologies may be applied to online learning systems to make the learning process more interactive. However, the majority of studies have been focusing on QA from a collection of text documents [4].

The overall aim of this QA track was to retrieve small pieces of text that contain the actual answer to the question rather than the list of documents traditionally returned by retrieval engines [Voorhees & Tice, 2000]. The TREC-8 QA track attracted researchers from both industry and academia. Twenty organizations participated in this track with different approaches and their systems were evaluated. The participating systems were tested on a huge set of unstructured documents and a set of fact-based questions.

Related work

To construct intellectual answering system, it must collect and store questions and answers organically. The intelligent Q&A system will contain a Q&A library, which is built through the semantic understanding of natural language technology [1].

- **Question Module:-**intelligent Q&A system supports a variety of issues, questions and retrieval style. The module will include some question types.

- ***The Answer Module:-***When students ask questions, the system will check keywords firstly, and then mark up the sentence. Then it will do retrieval according to keywords and phrase marks.
- ***The Database Management Module:-***The core of intelligent Q&A system is database management. This module can be divided into two sub module (1) classification of data storage management sub-module and (2)the classified data query sub-module.
- ***The Online Q&A Module:-***The Q&A on-line module will provide a multimedia support, or the simulation of a classroom.
- ***The Domain Knowledge Engine:-*** This module will use Web searching Engine to find answers from WWW.
- ***The System Management Module:-***This module is used by administrator to maintain the whole system, in order to level off its run.

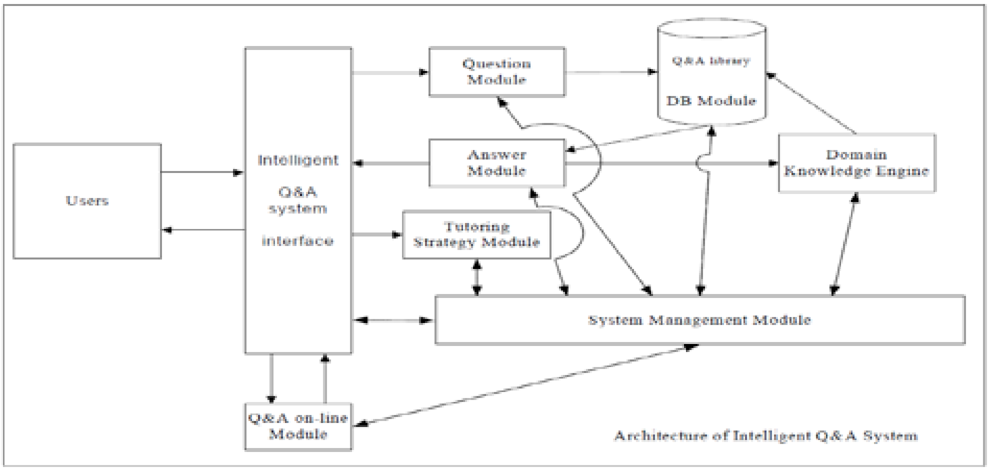


Fig. 1. Structure of Intelligent Q&A system.

How to generate the question from sentence. To generate the query question use the named Entity Recognition (NER) technology. Isolate meaningful sentences among the generated questions.

There are two types of generated questions.

(i)Sentence question

(ii)Entity question [2].

- Example:- Dr.Martin associated on April 4,1968 in Bhopal.
- Here Dr.Martin:<Person>
- April 4,1968<Time>
- Bhopal<Location>
- From above result “sentence question” for <Time> is “*when* was Dr. Martin associated in Bhopal”.
- *Entity question for* <Person> is *who is* Dr. Martin?

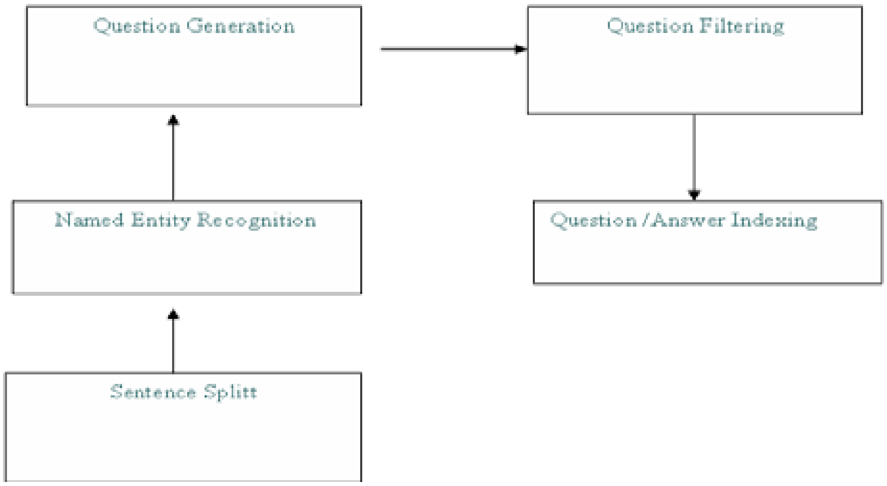


Fig. 2 Data flow of Question Answer System

Rextor (Relations EXtracTOR) is a document content analysis system designed to unify and generalize many previous natural language information retrieval techniques into one single framework. The system provides two separate grammars: one for extracting arbitrary entities from documents, and the other for building relations from the extracted items [3].

The Start System (Katz, 1990; Katz, 1997) analyzes English text and builds a knowledge base from information found in the text. START (SynTactic Analysis using Reversible Transformations) provides multimedia information access using natural language.

Intelligent Questing Answering Systems (IQASs) that are designed to interact meaningfully with, and adapt to, the users' input [8], Different (IQASs) use different Natural Language Processing (NLP) techniques in their system. NLP systems may be structural, i.e., focused on grammar and logic, or non-structural, i.e., focused on words and statistics.

Limitations:-

- Answers questions indirectly
- Does not attempt to understand the "meaning" of user's query or documents in the collection.

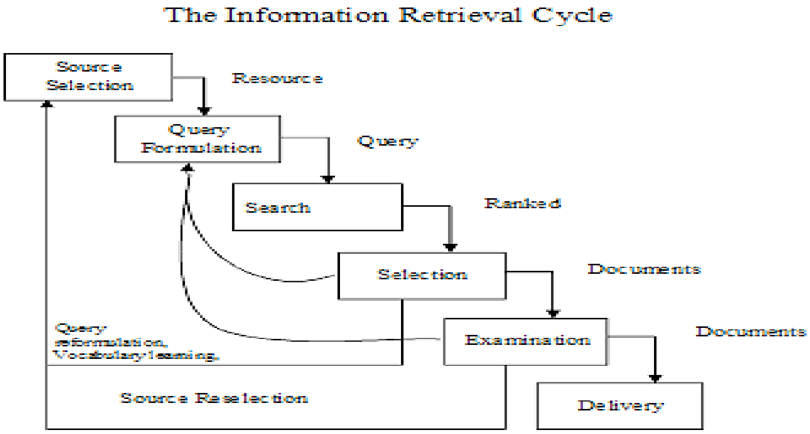
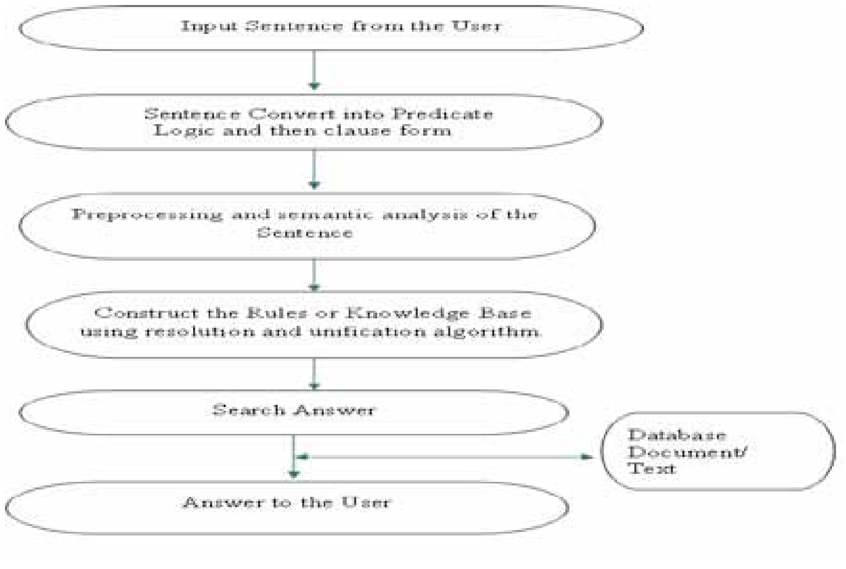


Fig. 3 Information Retrieval Cycle

Information Retrieval:

- Collection of documents (quantity)
- Set of information needs (topics)
- Sets of documents that satisfy the information needs (relevance judgments)
- Three components of a test collection:
- Exactness
- Recall
- Other measures derived there from

Proposed Framework


Input Sentence from the User
Sentence Convert into Predicate Logic and then clause form
Preprocessing and semantic analysis of the Sentence
Construct the Rules or Knowledge Base using resolution and unification algorithm
Search Answer
Database Document/ Text
Answer to the User

Proposed Algorithm:-The basic Steps in these algorithms are illustrated:-

1. Input: A Query as a Sentence.

2 Convert this sentence into predicate logic after then clause form.

3. Processing and semantics analysis of the sentence with the help of predicate logic and clause form. Now question is to be asked and converted into predicate logic and then clause form

.

4. Construct the rules or knowledge base using resolution and unification algorithm. Finally the clause form of the question is resolved to final answer using resolution and unification algorithm.

5. Similarity comparisons between input sentence and database of the text by using algorithm.

6. Finally relevant answer is retrieved with respect to corresponding query sentence.

7. Repeat step 1 to 7 for another query sentence.
8. End

Formalized Answer Extraction Based on Pattern Learning

In recent years, the combination of web growth, improvements in information technology, and the explosive demand for better information access have increased the interest in QA systems. Unlike most QA systems that face the problem of how to find short and correct answers to open-domain questions by searching a large collection of documents, this project is focused on finding patterns to formulate a “complete” and “natural” answer to questions, given the short answer. Finding such patterns is important as it can be used to enhance existing QA systems to provide answers to the user in a more “natural way”.

Answer extraction is typical two-category case, because one candidate answer only has two kinds of situations, whether it is an answer or not. Therefore, this kind of problem is suitable for the method of logistic regression for analyzing. But in the actual conditions, the positive instance (correct answer) far less than negative instance (interference answer), it brings about serious data sparse. In this case, if you directly adopt maximum-likelihood estimation, it will result in the model parameter and probability estimate deviation.

5.1 Algorithm to Convert input sentence into Predicate Logic

1) Separate all the words and store them in a string array or double-dimensional character array.

2) Use all the conjunctions to obtain subjects. E.g: "Ram" is subject, it is followed by "and" so whatever follows "and" will also be a subject, i.e. Syam, Raj.

3) Remove all the remaining articles. E.g: remove "a", "an" and "the"

4) Then combine all the other words

After that we apply the Unification algorithm. The unification problem in first-order logic can be expressed as follows: Given two terms containing some variables, find, if it exists, the ***simplest substitution*** (i.e., an assignment of some term to every variable) which makes the two terms equal. The resulting substitution is called the ***most general unifier.***

Algorithm: Unify (L1, L2)

```
1.  If $L1$ or $L2$ is a variable or constant, then:
        a) If $L1$ and $L2$ are identical, then return NIL.
        b) Else if $L1$ is a variable, then if $L1$ occurs in $L2$ then return FAIL, else return $\{(L2/L1)\}$.
        c) Else if $L2$ is a variable, then if $L2$ occurs in $L1$ then return FAIL, else return $\{(L1/L2)\}$.
        d) Else return FAIL.
```

```
2.  If the initial predicate symbols in L1 and L2 are not identical, then return FAIL.

3.  If L1 and L2 have a different number of arguments, then return FAIL

4.  Set SUBST to NIL.

5.  For $i \leftarrow 1$ to number of arguments in L1:
        a) Call Unify with the ith argument of L1 and the ith argument of L2, putting result in S.
        b) If S = FAIL then return FAIL.
        c) If S is not equal to NIL then:
            i). Apply S to the remainder of both L1 and L2.
            ii). SUBST: = APPEND(S, SUBST).

6.  Return SUBST.
```

Algorithm: Convert to Clause Form

```
1.  Eliminate→, using: $a \to b = \neg a \vee b$
2.  Reduce the scope of each ¬ to a single term, using:
        $\neg(\neg p) = p$
    De-Morgan's laws: -
        $\neg(a \wedge b) = \neg a \vee \neg b$
        $\neg(a \vee b) = \neg a \wedge \neg b$
        $\neg \forall x P(x) = \exists x \neg P(x)$
        $\neg \exists x P(x) = \forall x \neg P(x)$
3.  Standardize variables.
4.  Move all quantifiers to the left of the formula without changing their relative order.
5.  Eliminate existential quantifiers by inserting Skolem functions.
6.  Drop the prefix.
7.  Convert the expression into a conjunction of disjuncts, using associativity and distributivity.
8.  Create a separate clause for each conjunct.
9.  Standardize apart the variables in the set of clauses generated in step 8, using the fact that:
        $(\forall x : P(x) \wedge Q(x) = \forall x : P(x) \wedge \forall x : Q(x)$
10. End
```

5.2 The Basis of Resolution and Herbrand's Theorem

- Herbrand's Theorem:

Herbrand's theorem is a fundamental result of mathematical logic obtained by Jacques Herbrand (1930). It essentially allows a certain kind of reduction of first-order logic to propositional logic. Although Herbrand originally proved his theorem for arbitrary formulas of first-order logic, the simpler version shown here, restricted to formulas in

prenex form containing only existential quantifiers became more popular. This theorem also reduces the question of unsatisfiability in first-order logic to the question of unsatisfiability in propositional calculus.
To show that a set of clauses *S* is unsatisfiable, it is necessary to consider only interpretations over a particular set, called the *Herbrand universe* of *S*. A set of clauses *S* is unsatisfiable if and only if a finite subset of ground instances (in which all bound variables have had a value substituted for them) of *S* is unsatsifable.

Algorithm: This Algorithm is used for Resolution

1. Convert all the statements of *F* to clause form.

2. Negate *P* and convert the result to clause form. Add it to the set of clauses obtained in 1.

3. Repeat until either a contradiction is found, no progress can be made, or a predetermined amount of effort has been expended.

   a) Select two clauses. Call these the parent clauses.

   b) Resolve them together. The resolvent will be the disjunction of all the literals of both parent clauses with appropriate substitutions performed and with the following exception: If there is one pair of literals *T*1 and ¬ *T*2 such that one of the parent clauses contains *T*1 and the other contains ¬ *T*2 and if *T*1 and *T*2 are unifiable, then neither *T*1 nor ¬ *T*2 should appear in the resolvent. If there is more than one pair of complementary literals, only one pair should be omitted from the resolvent.

   c) If the resolving is the empty clause, then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

5.3 Example of sentence conversation and their answer

1. Who play the game?

Predicate logic: - Playgame (x, y)

Clause form: - Playgame(x, y)

Resolution: - ~ Playgame (x, y) 1

x = Ram and y = Syam

Answer: - Ram and Syam.

Conclusion

Natural language processing is a technique that includes both natural language understanding and natural language generation. Translating one natural language into another becomes complex due or structural difference, varieties of meanings, different forms of verbs etc. NLP is a very difficult task because human being has good common sense and reasoning mechanism which they use to answer the question. Had the task of NLP be easy then computer would have been told the story, ask questions and computer would have given answers and this preparation of the program would have required very less time. This paper is based on predicate logic for question answering system.

References

[1] Susu Jiang and Zhu Yonghua, “The Design for Semantic Based Intelligent Q&A System”, First International Workshop on Education Technology and Computer Science, IEEE,2009, pp.90-92.

[2] Min-kyoung kim and Han-joon Kim, “Design of question Answering System a with automated Question Generation”, IEEE, 2008, pp.365-369.

[3] Boris Katz and Jimmy Lin “REXTOR: A System for Generating Relations from Natural Language”,Proceeding of ACL workshop on NLP & IR 2000.

[4] Rich and Kelvin Knight, “Artificial Intelligence”, Mcgraw hill companies, chapter 15, pp.377-426.

[5] Yasunori Ishihara, Hiroyuki Seki and Tadao Kasami “A Translation Method from Natural Language Specifications into Formal Specifications Using Contextual Dependencies”,

[6] J. A. Goguen, J. W. Thatcher, and E. G. Wagner, “An initial algebra approach to the specification, correctness and implementation of abstract data types,” IBM Research Report, 1976

[7] Paeseler, H. Ney: “Continuous Speech Recognition Using a Stochastic Language Model” IEEE. Int. Conf. on Acoustics, Speech and Signal Processing, Glasgow, pp. 719-722, May 1989

[8] H. Seki, et al., “A Processing System for Program Specifications in a Natural Language,” Proc. 21th HICSS, pp. 754–763, Jan. 1988.

[9] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. “Question Answering passage retrieval using dependency relations”. In Proceedings of the 28th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pages 400–407, Salvador, Brazil, 2005.

[10] E. Sneiders, “Automated Question Answering: Template-Based Approach”, PhD thesis, Stockholm University / KTH press, Sweden, 2002.

[11] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, vol. 28, no. 1, pp. 11-21, 1972.

[12] CUI Heng, CAI Dongfeng, MIAO Xuelei□ “Research on Web-based Chinese Question Answering System and Answer Extraction”, Journal of Chinese Information Processing.2004,18(3):. Pp-24-31.

[13] James Allen “Natural Language Understanding”.

[14] S. Rajasekharan & G.A.Vijaya Lakshmi Pai, “Neural network, Fuzzy System and Genetic Algorithm”.

[15] Booch, G. (1994). “Object-Oriented Analysis and Design with Applications”, 2nd Ed., Benjamin Cummings.