Minimal parsing Key Concept based Question Answering System

Sunil Kopparapu[1], Akhlesh Srivastava[1], and P. V. S. Rao[2]

[1] Advanced Technology Applications Group, Tata Consultancy Services Limited, Subash Nagar, Unit 6 Pokhran Road No 2, Yantra Park, Thane West, 400 601, India.
[2] Tata Teleservices (Maharastra) Limited, B. G. Kher Marg, Worli, Mumbai, 400 018, India.

{sunilkumar.kopparapu, akhilesh.srivastava}@tcs.com, dr.pvs.rao@tatatel.co.in

Abstract. The home page of a company is an effective means for show casing their products and technology. Companies invest major effort, time and money in designing their web pages to enable their user's to access information they are looking for as quickly and as easily as possible. In spite of all these efforts, it is not uncommon for a user to spend a sizable amount of time trying to retrieve the particular information that he is looking for. Today, he has to go through several hyperlink clicks or manually search the pages displayed by the site search engine to get to the information that he is looking for. Much time gets wasted if the required information does not exist on that website. With websites being increasingly used as sources of information about companies and their products, there is need for a more convenient interface. In this paper we discuss a system based on a set of Natural Language Processing (NLP) techniques which addresses this problem. The system enables a user to ask for information from a particular website in free style natural English. The NLP based system is able to respond to the query by 'understanding' the intent of the query and then using this understanding to retrieve relevant information from its unstructured info-base or structured database for presenting it to the user. The interface is called UniqliQ as it avoids the user having to click through several hyperlinked pages. The core of UniqliQ is its ability to understand the question without formally parsing it. The system is based on identifying key-concepts and keywords and then using them to retrieve information. This approach enables UniqliQ framework to be used for different input languages with minimal architectural changes. Further, the key-concept – keyword approach gives the system an inherent ability to provide approximate answers in case the exact answers are not present in the information database.

Keywords: NL Interface, Question Answering System, Site search engine

1 Introduction

Web sites vary in the functions they perform but the baseline is dissemination of information. Companies invest significant effort, time and money in designing their web pages to enable their user's to access information that they are looking for as

quickly and as easily as possible. In spite of these efforts, it is not uncommon for a user to spend a sizable amount of time (hyperlink clicking and/or browsing) trying to retrieve the particular information that he is looking for. Until recently, web sites were a collection of disparate sections of information connected by hyperlinks. The user navigated through the pages by guessing and clicking the hyperlinks to get to the information of interest. More recently, there has been a tendency to provide site search engines[1] , usually based on key word search strategy, to help navigate through the disparate pages. The approach adopted is to give the user all the information he could possibly want about the company. The user then has to manually search through the information thrown back by the search engine i.e. search the search engine. If the hit list is huge or if no items are found a few times he will probably abandon the search and not use the facility again. According to a recent survey [1] 82 percent of users to Internet sites use on-site search engines. Ensuring that the search engine has an interface that delivers precise[2] , useful[3] and actionable[4] results for the user is critical to improving user satisfaction. In a web-browsing behavior study [7], it was found that none of the 60 participants (evenly distributed across gender, age and browsing experience) was able to complete all the 24 tasks assigned to them in a maximum of 5 minutes per task. In. that specific study, users were given a rather well designed home page and asked to find specific information on the site. They were not allowed to use the site search engine. Participants were given common tasks such as finding an annual report, a non-electronic gift certificate, the price of a woman's black belt or, more difficult, how to determine what size of clothes you should order for a man with specific dimensions.

To provide better user experience, a website should be able to accept queries in natural language and in response provide the user succinct information rather than (a) show all the (un)related information or (b) necessitate too many interactions in terms of hyperlink clicks. Additionally the user should be given some indication in case either the query is incomplete or an approximate answer in case no exact response is possible based on information available on the website. Experiments show that, irrespective of how well a website has been designed, on an average, a computer literate information seeker has to go through at least 4 clicks followed by a manual search of all the information retrieved by the search engine before he gets the information he is seeking[5] . For example, the Indian railway website [2], frequented by travelers, requires as many as nine hyperlink clicks to get information about availability of seats on trains for travel between two valid stations [9].

Question Answering (QA) systems [6][5][4], based on Natural Language Processing (NLP) techniques are capable of enhancing the user experience of the information seeker by eliminating the need for clicks and manual search on the part of the user. In effect, the system provides the answers in a single click. Systems using

[1] We will use the phrase "site search engine" and "search engine" interchangeably in this paper.

[2] In the sense that only the relevant information is displayed as against showing a full page of information which might contain the answer.

[3] In the absence of an exact answer the system should give alternatives, which are close to the exact answer in some intuitive sense.

[4] Information on how the search has been performed should be given to the user so that he is better equipped to query the system next time.

[5] provided of course that the information is actually present on the web pages

NLP are capable of understanding the intent of the query, in the semantic sense, and hence are able to fetch exact information related to the query.

In this paper, we describe a NLP based system framework which is capable of understanding and responding to questions posed in natural language. The system, built in-house, has been designed to give relevant information without parsing the query[6]. The system determines the key concept and the associated key words (KC-KW) from the query and uses them to fetch answers. This KC-KW framework (a) enables the system to fetch answers that are close to the query when exact answers are not present in the info-base and (b) gives it the ability to reuse the KC-KW framework architecture with minimal changes to work with other languages. In Section 2 we introduce QA systems and argue that neither the KW based system nor a full parsing system are ideal; each with its own limitations. We introduce our framework in Section 3 followed by a detailed description of our approach. We conclude in Section 4.

2 Question Answering Systems

Question Answering (QA) systems are being increasingly used for information retrieval in several areas. They are being proposed as 'intelligent' search engine that can act on a natural language query in contrast with the plain key word based search engines. The common goal of most of them is to (a) understand the query in natural language and (b) get a correct or an approximately correct answer in response to a query from a predefined info-base or a structured database.

In a very broad sense, a QA system can be thought of as being a pattern matching system. The query in its original form (as framed by the user) is preprocessed and parameterized and made available to the system in a form that can be used to match the answer paragraphs. It is assumed that the answer paragraphs have also been preprocessed and parameterized in a similar fashion. The process could be as simple as picking selective key words and/or key phrases from the query and then matching these with the selected key words and phrases extracted from the answer paragraphs. On the other hand it could be as complex as fully parsing the query[7] , to identify the parts of speech of each word in the query, and then matching the parsed information with fully parsed answer paragraphs. The preprocessing required would generally depend on the type of parameters being extracted. For instance, for a simple key words type of parameter extraction, the preprocessing would involve removal of all words that are not key words while for a full parsing system it could be retaining the punctuations and verifying the syntactic and semantic 'well-formedness' of the query.

Most QA systems resort to full parsing [4,5,6] to comprehend the query. While this has its advantages (it can determine who killed who in a sentence like "Rama killed Ravana") its performance is far from satisfactory in practice because for accurate and

[6] We look at all the words in the query as standalone entities and use a consistent and simple way of determining whether a word is a key-word or a key-concept.

[7] Most QA systems, available today, do a full parsing of the query to determine the intent of the query. A full parsing system in general evaluates the query for syntax (and followed by semantics) by determining explicitly the part of speech of each word

consistent parsing (a) the parser, used by the QA system and (b) the user writing the (query and answer paragraph) sentences should both follow the rules of grammar. If either of them fails, the QA system will not perform to satisfaction. While one can ensure that the parser follows the rules of grammar, it is impractical to ensure this from a casual user of the system. Unless the query is grammatically correct – the parser would run into problems. For example

- A full sentence parser would be unable to parse a grammatically incorrect constructed query and surmise the intent of the query[8] .
- Parsing need not always necessarily gives the correct or intended result. "Visiting relatives can be a nuisance to him", is a well known example[12], which can be parsed in different ways, namely, (a) visiting relatives is a nuisance to him. (him = visitor) or (b) visiting relatives are a nuisance to him. (him ≠ visitor).

Full parsing, we believe, is not appropriate for a QA system especially because we envisage the use of the system by

- large number of people who need not necessarily be grammatically correct all the time,
- people would wish to use casual/verbal grammar[9]

Our approach takes the middle path, neither too simple not too complex and avoids formal parsing.

3 Our Approach: UniqliQ

UniqliQ is a web enabled, state of the art intelligent question answering system capable of understanding and responding to questions posed to it in natural English. UniqliQ is driven by a set of core Natural Language Processing (NLP) modules. The system has been designed keeping in mind that the average user visiting any web site works with the following constraints

- the user has little time, and doesn't want to be constrained by how he can or can not ask for information[10]
- the user is not grammatically correct all the time (would tend to use transactional grammar)
- a first time user is unlikely to be aware of the organization of the web pages
- the user knows what he wants and would like to query as he would query any other human in natural English language.

Additionally, the system should

- be configurable to work with input in different languages
- provide information that is close to that being sought in the absence of an exact answer
- allow for typos and misspelt words

---

[8] the system assumes that the query is grammatically correct

[9] intent is conveyed; but from a purist angle the sentence construct is not correct

[10] In several systems it is important to construct a query is a particular format. In many SMS based information retrieval system there is a 3 alphabet code that has to be appended at the beginning of the query in addition to sending the KWs in a specific order.

The front end of UniqliQ, shown in Fig. 1, is a question box on the web page of a website. The user can type his question in natural English. In response to the query, the system picks up specific paragraphs which are relevant to the query and displays them to the user.

3.1 Key Concept-Key Word (KC-KW) Approach

The goal of our QA system is (a) to get a correct or an approximate answer in response to a query and (b) not to put any constraint on the user to construct syntactically correct queries[11] . There is no one strategy envisaged – we believe a combination of strategies based on heuristics, would work best for a practical QA system. The proposed QA system follows a middle path especially because the first approach (picking up key words) is simplistic and could give rise to a large number of irrelevant answers (high false acceptances), the full parsing approach is complex, time consuming and could end up rejecting valid answers (false rejection), especially if the query is not well formed syntactically. The system is based on two types of parameters -- key words (KW) and key concepts (KC).

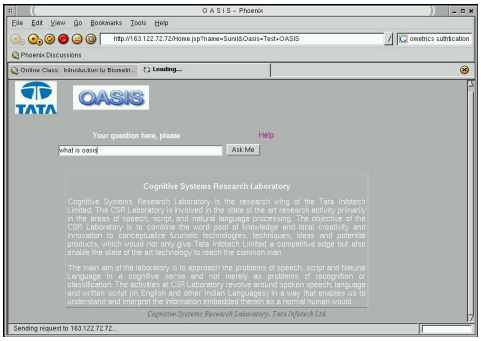

Fig. 1. Screen Shot of UniqliQ system

In each sentence, there is usually one word, knowing which the nature of these semantic relationships can be determined. In the sentence, "I purchased a pen from Amazon for Rs. 250 yesterday" the crucial word is 'purchase'. Consider the expression, Purchase(I, pen, Amazon, Rs. 250/-, yesterday). It is possible to understand the meaning even in this form. Similarly, the sentence "I shall be traveling to Delhi by Air on Monday at 9 am" implies: Travel (I, Delhi, air, Monday, 9am). In the above examples, the key concept word 'holds' or 'binds' all the other key words together. If the key concept word is removed, all the others fall apart. Once the key concept is known, one also knows what other key words to expect; the relevant key

[11] Verbal communication (especially if one thinks of a speech interface to the QA system) uses informal grammar and most of the QA systems which use full parsing would fail.

words can be extracted. There are various ways in which key concepts can be looked at

1. as a mathematical functional which links other words (mostly KWs) to itself. Key Concepts are broadly like 'function names' which carry 'arguments' with them. E.g. KC1 (KW1, KW2, KC2 (KW3, KW4))
   Given the key concept, the nature and dimensionality of the associated key words get specified.

   We define the arguments in terms of syntacto-semantic variables: e.g. destination has to be "noun – city name"; price has to be "noun – number" etc.
   Mass-of-a-sheet (length, breadth, thickness, density)
   Purchase (purchaser, object, seller, price, time)
   Travel (traveler, destination, mode, day, time)
2. as a template specifier: if the key concept is purchase/sell, the key words will be material, quantity, rate, discount, supplier etc. Valence, or the number of arguments that the key concept supports is known once the key concept is identified.
3. as a database structure specifier: consider the sentence, "John travels on July 20th at 7pm by train to Delhi". The underlying database structure would be

KeyCon	KW1	KW 2	KW3	KW4	KW5
Travel	Traveler	Destination	Mode	Day	Time
	John	Delhi	Train	July_20	7 pm

KCs together with KWs help in capturing the total intent of the query. This results in constraining the search and making the query very specific. For example, reserve (place_from = Mumbai, place to=Bangalore, class=2nd), makes the query more specific or exact, ruling out the possibility of a reservation between Mumbai and Bangalore in 3rd AC for instance.

A key concept and key word based approach can be quite effective solution to the problem of natural (spoken) language understanding in a wide variety of situations, particularly in man-machine interaction systems.

The concept of KC gives UniqliQ a significant edge over simplistic QA systems which are based on KWs only [3]. Identifying KCs helps in better understanding the query and hence the system is able to answer the query more appropriately. A query in all likelihood will have but one KC but this need not be true with the KCs in the paragraph. If more than one key concept is present in a paragraph, one talks of hierarchy of key concepts[12] . In this paper we will assume that there is only one KC in an answer paragraph.

One can think of a QA system based on KC and KW as one that would save the need to fully parse the query; this comes at a cost, namely, this could result in the system not being able to distinguish who killed whom in the sentence "Rama killed Ravana". The KC-KW based QA system would represent it as kill (Rama, Ravana) which can have two interpretations. But in general, this is not a huge issue unless

[12] when several KCs are present in the paragraph then one KC is determined to be more important than another KC

there are two different paragraphs – the first paragraph describing about Rama killing Ravana and a second paragraph (very unlikely) describing Ravana killing Rama.

There are reasons to believe that humans resort to a key concept type of approach in processing word strings or sentences exchanged in bilateral, oral interactions of a transactional type. A clerk sitting at an enquiry counter at a railway station does not carefully parse the questions that passengers ask him. That is how he is able to deal with incomplete and ungrammatical queries. In fact, he would have some difficulty in dealing with long and complex sentences even if they are grammatical.

3.2 Description

UniqliQ has several individual modules as shown in Fig. 2. The system is driven by a question understanding module (see Fig. 2). (Its first task as in any QA system is preprocessing of the query: (a) removal of stop words and (b) spell checking.) This module not only identifies the intent of the question (by determining the KC in the query) and checks the dimensionality syntax[13] [14] . The intent of the question (the key concept) is sent to the query generation module along with the keywords in the query.

The query module, assisted by a taxonomy tree, uses the information supplied by the question understanding module to specifically pick relevant paragraphs from within the website. All paragraphs of information picked up by the query module as being appropriate to the query are then ranked[15] in the decreasing order of relevance to the query. The highest ranked paragraph is then displayed to the user along with a context dependent prelude to the user. In the event an appropriate answer does not exist in the info-base, the query module fetches information most similar (in a semantic sense) to the information sought by the user. Such answers are prefixed by "You were looking for ....., but I have found ... for you" which is generated by the prelude generating module indicative that the exact information is unavailable. UniqliQ has memory in the sense that it can retain context information through the session. This enables UniqliQ to 'complete' a query (in case the query is incomplete) using the KC-KW pertaining to previous queries as reference.

At the heart of the system are the taxonomy tree and the information paragraphs (info-let). These are fine tuned to suit a particular domain. The taxonomy tree is essentially a word-net [13] type of structure which captures the relationships between different words. Typically, relationships such as synonym, type_of, part_of are captured[16] . The info-let is the knowledge bank (info-base) of the system. As of now,

---

[13] Dimensionality syntax check is performed by checking if a particular KC has KWs corresponding to an expected dimensionality. For example in a railway transaction scenario the KC reserve should be accompanied by 4 KWs where one KW had the dimensionality of class of travel, 1 KW has the dimensionality of date and 2 KWs have the dimensionality of location.

[14] The dimensionality syntax check enables the system to quiz the user and enable the user to frame the question appropriately

[15] Ranking is based on a notional distance between the KC-KW pattern of the query and the KC-KW pattern of the answer paragraph.

[16] A taxonomy is built by first identifying words (statistical n-gram (n=3) analysis of words) and then manually defining the relationship between these selected words. Additionally the

it is manually engineered from the information available on the web site[17] . The info-base essentially consists of a set of info-lets. In future it is proposed to automate this process.

The no parsing aspect of UniqliQ architecture gives it the ability to operate in a different language (say Hindi) by just using a Hindi to English word dictionary[18] . A Hindi front end has been developed and demonstrated [9] for a natural language railway enquiry application. A second system which answers agriculture related questions in Hindi has also been implemented.

3.3 Examples

UniqliQ platform has been used in several applications. Specifically, it has been used to disseminate information from a corporate website, a technical book, a fitness book, yellow pages[19] information retrieval [11] and railway [9]/ airline information retrieval. UniqliQ is capable of addressing queries seeking information of various types.

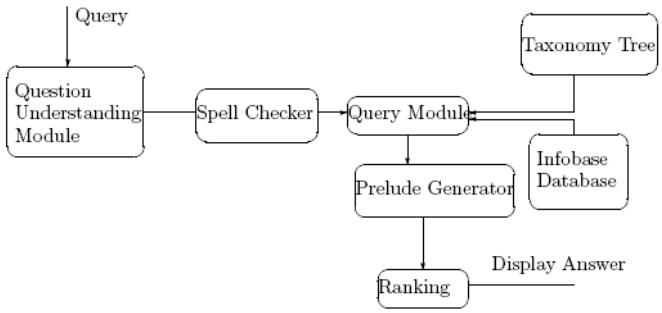

Fig. 2. The UniqliQ system. The database and info-base contain the content on the home page of the company.

Fig 3 captures the essential differences between the current search methods and the system using NLP in the context of a query related to an airline website. To find an answer to the question, ”Is there a flight from Chicago or Seattle to London?” on a typical airline website, a user has first to query the website for information about all the flights from Chicago to London and then again query the website to seek information on all the flights from Seattle to London. UniqliQ can do this in one shot and display all the flights from Chicago or Seattle to London (see Fig. 3). Fig. 4 and

selected words are tagged as key-words, key-concepts based on human intelligence (common sense and general understanding of the domain)

[17] A infolet is more often a paragraph which is self contained and ideally talks about a single theme

[18] Traditionally one would need a automatic language translator from Hindi to English

[19] User can retrieve yellow pages information on the mobile phone. The user can send a free form text as the query (either as an SMS or through a BREW application on a CDMA phone) and receive answers on his phone

Fig. 5 capture some of the questions the KC-KW based system is typically able to deal with.

The query ”What are the facilities for passengers with restricted mobility?” today typically require a user to first click the navigation bar related to Products and services; then search for a link, say, On ground Services; browse through all the information on that page and then pick out relevant information manually. UniqliQ it is capable of picking up and displaying only the relevant paragraph, saving time of the user also saving the user the pain of wading through irrelevant information to locate the specific item that he is looking for!

Ramesh: *Are there any flights from Chicago to London between 6 p.m. and 9 p.m. on Thursday?*
Shyam: *Can you please give me all the flights from Chicago to London which start from Chicago between 6 and 9 PM on Thursday?*

Currently, Ramesh and Shyam have to go the Airline *website ... select the source (Chicago) and destination (London) cities from a drop down menu ... the web page displays all the Chicago-London flights ... Then Ramesh and Shyam have to manually scan the displayed list to get the information they seek. A laborious process ...*
Uniqlic: Displays only the information of flights from Chicago to London which are functional on a Thursday between 6 and 9 PM.

Fig. 3. A typical session showing the usefulness of a NLP based information seeking tool against the current information seeking procedure.

- Is there a flight from Seattle to Chicago before noon?
- Give me the details of S2-112?
- Please give me the flights from Chicago to London which don't fly on Wednesday?
- Are there flights from Chicago to London on 25/9/2003?

Fig. 4. Some queries that UniqliQ can handle and save the user time and effort (reduced number of clicks)

- Can I carry my pets on my flight from Chicago to London?
- I am 65 years old. Do you provide any concession?
- Would you provide assistance for my 2 years old kid to travel alone?

Fig. 5. General queries that UniqliQ can handle and save the user manual search.

4. Conclusions

Experience shows that it is not possible for an average user to get information from a web site with out having to go through several clicks and manual search. Conventional site search engines lack the ability to understand the intent of the query; they operate based on keywords and hence flush out information which might not be useful to the user. Quite often the user needs to manually search amongst the search engine results for the actual information he needs. NLP techniques are capable of making information retrieval easy and purposeful. This paper describes a platform which is capable of making information retrieval human friendly. UniqliQ built on

NLP technology enables a user to pose a query in natural language. In addition it takes away the laborious job of manually clicking several tabs and manual search by presenting succinct information to the user. The basic idea behind UniqliQ is to enable a first time user to a web page to obtain information without having to surf the web site. The question understanding is based on identification of KC-KW which facilitates using the platform usable for queries in different languages. It also helps in ascertaining if the query has all the information needed to give an answer. The KC-KW approach allows the user to be slack in terms of grammar and works well even for casual communication. The absence of a full sentence parser is an advantage and not a constraint in well delimited domains (such as homepages of a company). Recalling the template specifier interpretation of key concept, it is easy to identify in case any required key word is missing from the query; e.g. if the KC is purchase/sell, the system can check and ask if any of the requisite key words (material, quantity, rate, discount, supplier) is missing. This is not possible with systems based on key words alone.

Ambiguities can arise if more than one key words have the same dimensionality (i.e. belong to the same syntacto-semantic category). For instance, the key concept 'kill' has: killer, victim, time, place etc. for key words. Confusion is possible between killer and victim because both have the same 'dimension' (name of human), e.g. kill who Oswald? (Who did Oswald kill - Kennedy, or who killed Oswald? - Jack Ruby)

Acknowledgments. Our thanks are due to members of the Cognitive Systems Research Laboratory. Several of whom have been involved in developing prototypes to test UniqliQ, the question answering system in various domains.

References

1. http://www.coremetrics.com/solutions/on_site_search.html
2. Indian Rail. http://www.indianrail.gov.in.
3. Eugene Agichtein, Steve Lawrence, and Luis Gravano. Learning search engine specific query transformations for question answering. In Proceedings of the Tenth International World Wide Web Conference. 2001.
4. AskJeevs. http://www.ask.com
5. AnswerBug. http://www.answerbug.com.
6. START. http://start.csail.mit.edu/.
7. WebCriteria. http://www.webcriteria.com.
8. Sunil Kopparapu, Akhilesh Srivastava, PVS Rao KisanMitra: A Question Answering System For Rural Indian Farmers, International Conference on Emerging Applications of IT (EAIT 2006) Science City Kolkata, February 10-11, 2006.
9. Sunil Kopparapu, Akhilesh Srivastava, PVS Rao Building a Natural Language Interface for the Indian Railway website, NCIICT 06, Coimbatore, July 7-8, 2006.
10. Sunil Koparapu, Akhilesh Srivastava, PVS Rao, Succinct Information Retrieval from Web, Whitepaper, Tata Infotech Limited (now Tata Consultancy Services Limited), 2004.
11. S Kopparapu, A Srivastava, S Das, R Sinha, M Orkey, V Gupta, J Maheswary, PVS Rao Accessing Yellow Pages Directory Intelligently on a Mobile Phone Using SMS, MobiComNet 2004 Vellore
12. http://www.people.fas.harvard.edu/~ctjhuang/lecture_notes/lecch1.html
13. http://wordnet.princeton.edu/