Buliding a Natural Language Interface for a Railway Website

Sunil Kopparapu, Akhilesh Srivastava, PVS Rao
Cognitive Systems Research Laboratory
Tata Consultancy Services Limited, Mumbai
sunilkumar.kopparapu@tcs.com

***Abstract*— Rail is the most widely used mode of transport in India today. Everyday millions of passengers travel in thousands of trains from one corner of India to another. The usual practice is to reserve accommodation in advance before travel. The traveler has plenty of options in the form of different trains, grades (like sleeper, II AC, chair-car) in addition to the date of travel. More recently, this information is being made available online by the Indian railway [1] which has scores of travelers flocking to, to get information. Fetching information is tedious and requires selection from several drop down menus, several hyperlink clicks and manual browsing to get to the desired information. In this paper, we propose a Natural Language Interface (NLI) to the Indian railway website. NLI enables easy and succinct information retrieval by posing queries in natural English.**

I. Background

Search engines such as AltaVista [2] and Google [3] typically treat a natural language question as a list of keywords and retrieve documents that have *similar* keywords. However, documents with the best answers may not contain all the keywords in the original query and hence may be ranked low by the search engine. These queries could be answered more precisely if a search engine recognized them as questions as humans would do and not merely treat the queries as a set of keywords [4]. To cater to the growing world wide web and the need to search by the semantic intent of the query a large number of QA systems have been researched, eg [5], [6], [7].

In a study conducted [8] to study the web-browsing behavior, it was found that none of the 60 participants equally divided into groups representing gender, age and browsing experience was able to complete all the tasks assigned to them in a specified period of time. In that specific study, users were given a rather well designed home page and specific information to find on a site. They were not allowed to use the site conventional search engine. Participants were asked to complete 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". They were given a maximum of five minutes per task and a total of 24 tasks. None of the participants completed all 24 tasks in the allotted time [9]. To provide a better user experience the web site should be able to provide him with succinct information rather than show him all the information! In addition he should be given some expected answer in case he doesn't provide all information in his query. Today, on an average, a computer literate information seeker has to go through at least 4 clicks and manual search before he gets to the information he is seeking, irrespective of how well a web page has been designed; provided of course that the information is actually present on the web pages.

The Indian Railway website [1] is official and authoritative resource of information pertaining to Indian Rail. The website houses information pertaining to several trains traveling through thousands of railway station in addition to holding information about status of accommodation in each train between every station it passes through. This information is viewed by millions of travelers to reserve tickets in advance for their journey and also to the check the status of their wait listed ticket. Even for an average computer literate it is quite tedious and time consuming to obtain information from any website, the Indian Railway website is not exception. In the Indian Railway website scenario, to get any information, invariably he has to select from several drop down menus, navigate through several hyperlinks and then manually browse; the task gets even more involved if he is not sure about the name/number of trains traveling between the desired stations.

Question Answering (QA) system based on Natural Language Processing (NLP) techniques are capable of enhancing the user experience of the information seeker by removing the labor of hyperlink clicks and manual search from the user. In effect, the system emulates the several clicks and manual search performed by the user. Systems using NLP are capable of understanding the intent of the query, in the semantic sense, and hence are able to fetch exact and crisp information related to the query.

In this paper we describe an *intelligent* and *effective* interface, that enables retrieval of information from [1] using natural English both easy and convenient. The system, is an NLI to [1] and has the following salient features

- It is easy and a natural interface to use
- does not require the user to remember any specific station code or train number, just ask for what (s)he wants
- system assumes the most commonly used values for the absent parameters in case of an incomplete user query.

II. The Problem

Searching for information on the web is very tedious [9] and could get worse if one is in a hurry to get that information. For example let us consider a situation, *Ram* is currently in

Ramesh**:** *Is there a seat available on December 25 from Mumbai to Delhiin III AC?*
Sita**:** *Can you please give me all the trains from Mumbai to Delhi on which I can reserve a ticket for December 25?*

Currently, Ramesh *and* Sita *have to go the Indian Railway website ... type in the source (Mumbai) and destination (Delhi) .. then from a drop down menu select the exact station code plus the date of travel ... the web page then displays all the Mumbai-Delhi trains ... Then* Ramesh *and* Sita *have to then select a train (from the presented list) and then click to get information about the seat availability in that train. A laborious process ... several hyperlink clicks and choices from drop down menus.*

NL System: Displays accommodation information in all trains from Mumbai to Delhias a table. All this in one hyperlink click.

Fig. 1. A typical session showing the usefulness of a NLP based information seeking tool against the current information seeking procedure.

Mumbai and he needs to travel to Delhi on, say, 30th July and he wants to check the ticket availability. Further, he does not know details of any specific trains that commute between Mumbai and Delhi.

Today, to fetch this information from the Indian railway website [1] would require him to specify the source and destination (spelled correctly) in pre specified text boxes and then click. After a finite wait the system extracts the station codes of the source and destination stations if they are unique, else one will need to select the desired source and destination stations from the list provided. Post selection of the source and destination, one would have to choose from a list of grades of travel. This results in the system presenting a list of trains that run between the source and the destination stations. Each train in the presented list can be selected one by one and a hyperlink clicked to check for the availability of accommodation in an individual train. For example, if there are N trains between the chosen source and destination, in the worst case scenario one would have typically clicked $N + 3$ hyperlinks and chosen N selections. This aspect of search for information is both tedious and time consuming.

An NL based user interface would make it much simpler and faster and can be completed in one hyperlink click. Figure 1 captures the information retrieval from [1] as it done today and the user experience an NL system can contribute.

In general, the information that is *directly* accessible from the railway website [1] are

1) Status of a already reserved ticket based on PNR number
2) List of trains between a pair of station,
3) Availability of seat in a particular class on a particular train and on a particular date.
4) Schedule of a specific train
5) Cost of travel by a particular train, particular class.

Fig. 2. Screen Shot of the natural English interface to the Indian Railway website [1].

While the user additionally (*indirect query*[1]) is looking for

1) In which train (between two stations) a seat is available in a range of dates,
2) Does a particular train stops as a particular station enroute
3) When is the earliest reservation available between two stations?
4) PNR status of his to and fro tickets.

In this paper, we present an NL interface (see Figure 2) which while giving the user access to the services provided by railway website [1] makes it convenient, saves time, minimizes hyperlink clicks and manual browsing. In addition the NL system extracts and presents the actual information one is looking for. The system is web enabled, uses the state of the art intelligent question answering system capable of understanding and responding to questions posed to it in natural English.

III. System Overview

The input to the NL system (essentially a QA system) is a user query, typed in a text box (Figure 2) in natural English and the final output is the data dynamically retrieved from [1] which is parsed and formatted to present the retrieved data meaningfully. The system has the capability of correcting spelling mistakes and does not require the user to construct grammatically correct sentences. The system can be logically depicted as being made up of several modules (Figure 3).

- Question analyzing module: The input to this module is the typed user query in natural English, This module derive the intent of the query using NLP techniques by extracting *key concepts* and *key parameters* from the posed query. The module is able to establish the query intent by shallow parsing the query. The module is assisted by a database containing morphological equivalents, abbreviations and a spell checker. The intent of the query is expressed as a set of queries that need to be done on [1]. For the query, Is there a seat available

[1]Query requiring multiple accessing of the railway website

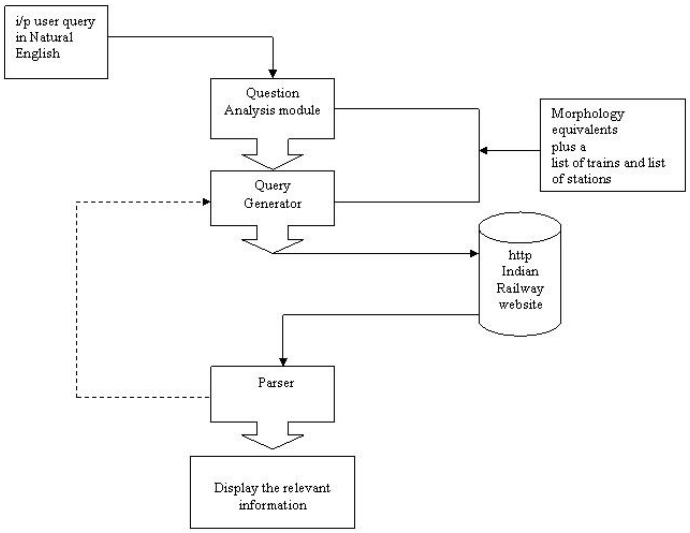

Fig. 3. Block diagram of the overall system to query Indian Railway

on December 25 from Mumbai to Delhi in III AC? the module establishes that the following sequential queries that need to be performed

1) find list of trains between Mumbai and Delhi
2) for each train and for default grade of travel for December 25 get status of seat available

- Query generating module: There is a predetermined format, in which [1] expects to be queried. The raw queries formed by the earlier module is transformed into a http query with proper syntax that can be passed to the Indian railway website [1]. Query `/cgi_bin/inet_accavl_cgi.cgi` with the key parameters, `lccp_trnno = XYZ, lccp_srccode = Mumbai, lccp_dstncode = Delhi, lccp_month = 12, lccp_day = 25, lccp_year = 2005, lccp_quota = GN, lccp_classopt = SL`. to get the seats available in train number XYZ. Note that XYZ is derived from an earlier query on list of trains between Mumbai and Delhi, namely, query `/cgi_bin/inet_srcdest_cgi.cgi` with `lccp_classopt = SL, lccp_src_stncode = Mumbai, lccp_dstn_stncode = Delhi`.
- Retrieving information, Parse and Display: The generated query is passed to the railway site and the result is retrieved. In case of direct queries, the response from [1] is minimally parsed while in case of *indirect* queries a more complex parsing is to be carried out to display information gathered from multiple http requests.

IV. Scope of NL Interface

The proposed system can handle variations in queries, additionally, it also enables the user to obtained information which is not directly serviceable on the Indian railway website. The examples below capture the scope of the NL interface.

Case 1: Query self contained and has all the required information to obtain information from the Indian railway website.

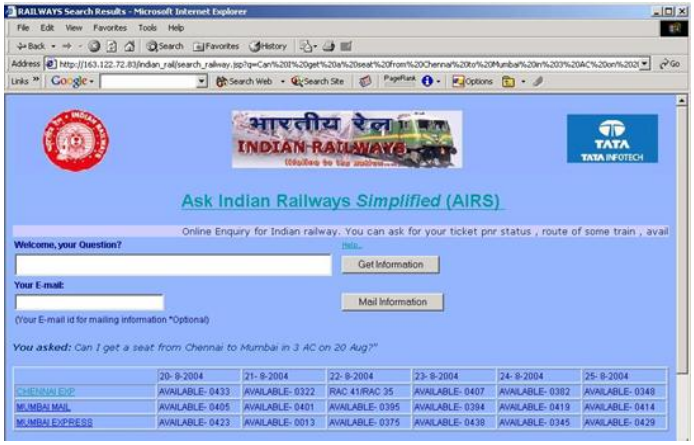

Fig. 4. Response to Can I get a seat from Chennai to Mumbai in 3 AC on 20 Aug?

Example:

1) What is status of my ticket bearing PNR number 9877387678?
2) What is the schedule of train number 2724
3) Which trains runs between Mumbai and Delhi
4) What is the fare from Mumbai to Delhi in train number 2905 in III AC?
5) Can I get a seat in 3 ac in train number 2905 on 25 July from Delhi to Mumbai.

The system can process these queries by generating an appropriate search query to obtain information from [1].

Case 2: Query does not contain all the information required to make a search on the railway website. For example:

1) Can I get a seat in train number 2905 on 25 July from Delhi to Mumbai.
   In this case there is no mention of the class of travel. The system appends sleeper class to the query and searches, alternatively the system could be configured to get information for all classes of travel.
2) Can I get a seat in train number 2905 in the month of July from Delhi to Mumbai.
   Here, there is no mention of a specific date of travel, In such cases the system supplements the date parameter using the following rule:
   - *If current month is not the month specified in query: The system gives out seat availability for the month specified in query.*
   - *The current month and month specified in query are same: The system gives information for today.*
   - *Specified month come prior to the current month: The system gives information for the specified month in the next year.*

Case 3: Query is incomplete and can not be processed without knowing some specific information which can not be commonly assumed or generated. Example (see Figure 4)

1) Can I get a seat on 20 July from Mumbai to Delhi in 2 AC?

Good Answers	87.61
Incorrect Answers	12.39

TABLE I

PERFORMANCE OF THE SYSTEM

In this case, it is assumed that the user is interesting in getting ticket from Mumbai to Delhi in 2 AC, he does not care for which specific train. The system gives the availability status in all the trains between Mumbai and Delhi.

2) What is the fare from Mumbai to Delhi in 2 AC?
The system gives the fare from Mumbai to Delhi in all the trains running between Mumbai and Delhi.

V. RESULT

The system was put to test by 8 users, who were not associated with the development of the system, but had used the Indian Railway website to get information and check status of tickets etc.

In a preliminary test, the users were themselves asked to rate the response into one of the categories (Good Answers and Incorrect Answers). In all 105 questions were fired to the system. The performance of the system is tabulated in Table I.

Response time for a user query significantly depends on the Internet connectivity of Indian Railway website from the place the system is housed at the time of firing the query. Though the response time depends on the nature of query (Especially Case 3 and 4 queries, where one would have to connect to the Indian railway website multiple times to get information information) the average time to get response from different type of query was found to be 13.5 seconds.

VI. CONCLUSION

Inspite of the effort and money spent on designing good websites the phenomenon of being able to get the information that one is seeking quickly and effortlessly is a challenge. In this paper, we have proposed an NL interface which allows querying the Indian railway in simple and natural English language. Several NLP techniques were used to process and analyze the query to understand the intent of the query and then retrieve the desired information dynamically from the Indian Railway website. The obtained information is then parsed to enable display of information for easy consumption. The system can

- handle a wide variety of variations in the query due to its shallow parsing which extracts key concept and key parameters.
- in the absence of complete information required to query the railway website, the system smartly adds the missing information that are most likely and fetches information. This is quite similar to how a human would respond to a question when he does not have the full information about it.

The developed interface makes it easy to query the system, even for a naive computer user. The user types his query in natural English and the system traverses the hyperlinks, browses for information and displays only the specific and crisp information.

REFERENCES

[1] I. Rail, in *http://www.indianrail.gov.in*, website.
[2] AltaVista, in *http://www.altavista.com*, website.
[3] Google, in *http://www.google.com*, website.
[4] E. Agichtein, S. Lawrence, and L. Gravano, "Learning search engine specific query transformations for question answering," in *Proceedings of the Tenth International World Wide Web Conference*, 2001.
[5] AskJeevs, in *http://www.ask.com*, website.
[6] AnswerBug, in *http://www.answerbug.com*, website.
[7] START, in *http://start.csail.mit.edu/*, website.
[8] WebCriteria, in *http://www.webcriteria.com*, website.
[9] C. Conrath, in *Creating a Web site that isn't a web, http://www.itworldcanada.com/a/ComputerWorld/*, website.