SMS based Natural Language Interface to Yellow Pages Directory

Sunil Kumar Kopparapu*, Akhilesh Srivastava, Arun Pande
TCS Innovation Labs - Mumbai
Tata Consultancy Services Limited
Yantra Park, Thane (West) - 400 601.
{SunilKumar.Kopparapu, Akhilesh.Srivastava, Arun.Pande}@TCS.Com

ABSTRACT

Yellow Pages are directories that source information about various commercial organizations like their addresses, phone contact and other details. These are very useful and are used by individual and other business houses. Until recently, the only way to access these yellow pages directory information was to physically look into a huge hard-copy directory, which was not only laborious but also time consuming and required the user to be familiar with the organization of the directory. More recently, there have been IVR based contact centers that have been set up which can be used by the users to query information. While it is easier than browsing through the physical directory, it still has several pitfalls. The time spent on trying to get the information is quite large and at the end of enquiry one is not sure if one will get the information that one is looking for. In this paper, we propose a novel interface which enables accessing the yellow pages directory information on the mobile phone by sending a short message service (SMS). The central idea of the proposed method is to avoid any constraint on the way the user can query the yellow pages directory except that it be in natural English. The system, which uses natural language processing (NLP) techniques, understands the intent of the query and intelligently searches the yellow pages directory to retrieve information. This retrieved information is then sent back to the user in the form of a SMS.

1. BACKGROUND

Mobile phones have made significant inroad into the society in the last couple of years. As we write, there is a large population that is going mobile. While the competition is on for the mobile service provider, one of the ways to retain current subscribers and attract new subscribers is to provide them with value added services (VAS) and at the same time increase the average revenue per user (ARPU). The mobile service provider can retain and increase the ARPU if they provide their subscribers application that are not only innovative but also useful in day to day life[3]. In India, like most of the developing countries, there is a trend to use SMS more than the voice because of economic sense.

Yellow pages directory is a very useful information resource that houses information about commercial organizations. It is very common for a directory to be available for every town or city and it is very often used to get information about the companies. Until recently, the physical yellow pages directory was the only source of information. To get information a user browsed through the directory and got to the information that he was looking for through the index. There is always the problem of how easy or difficult to reach the information is depending on the organization of the yellow pages directory. Unless very familiar, a user would take effort and time to get to the information. In the recent past yellow pages directories have become accessible through interactive voice response (IVR) systems [4][1]. The user is convenienced in the sense that he just needs to make a phone call and request for information; a live agent would search the yellow pages database directory using a series of SQL queries and convey the information back on the phone to the user. While it is a simpler solution, it is

(a) Time consuming (very often one has to be in the queue listening to advertisement or a very irritation "Please be on hold, your call is important to us and we will get back to you as soon as one of our agents is free to take your call"),

(b) Expensive for the user (telephone bill for the whole duration of the call including the time taken to get to the live agent!)

(c) Expensive for the service provider (need to set up a call center and engage 24×7 live agents) and

(d) Highly dependent on the searching skill of the live agent.

In addition acoustic confusion prevail which prolong the interactive session between the user and the live agent because of

(i) the noisy and low bandwidth telephonic conversation and

*Corresponding author

(ii) pronunciation and accent.

In this paper we describe an intelligent and effective system, on the mobile network using SMS that enables searching the yellow pages directory using natural English. The system proposed in this paper has the following salient points:

(a) System should be easy to use

(b) Should not require the user to remember any specific code or pneumonic to query the directory

(c) In the absence of exact information (in the database) in response to the query, the system should provide the user with next close answer in some sense

(d) Should cater to SMS lingo and typographic errors that might occur when generating an SMS.

2. SYSTEM OVERVIEW

The system consists of essentially three main modules (see Figure 1). The first module interfaces with the SMS gateway of the telecom operator and passes the SMS query to the second module, which is the heart of the system. This module which is based on the natural language processing techniques analyzes the query in natural English and generates a set of database queries and passes it on to the next module which is the database query module. The output of the query module is passed back to the interface module, which sends back the retrieved yellow pages information back to the user as an SMS. Figure 1 gives the overview of the system that is able to intelligently access the yellow pages directory on the mobile network using SMS.

The SMS module interfaces with the SMSC (SMS center) of the mobile service provider. Its main functionality is to obtain the SMS query from the SMSC and pass on the query to the NLP module, which is the heart of the system. The NLP module understands the intent of the query rather than looking at the query for predetermined set of keywords and generates a list of possible search criteria which can be used to generate an SQL query to extract yellow pages listings from the database. The first search criteria would essentially be what a plain search engine would do, namely, to use the words in the query itself to search the database, but the subsequent search criteria depends on the actual query itself and can be considered as dilating the constraints placed on the search. The dilation happens in a manner so that not all dimension of the query are diluted simultaneously. We will discuss this in more detail in a later section of this paper.

The system is designed such that it can be easily configured to interface with a SMSC. The SMS interface gets the SMS query and sends it as it is to the NLP module.

The NLP module initially tags each of the words in the query to belong to either the name of the company or the name of a place or a search word (using its knowledge base). In the event of a word not being tagged because of lack of knowledge; it is tagged as unknown and checked for possible spelling mistake using the spell checker module and then tagged appropriately.

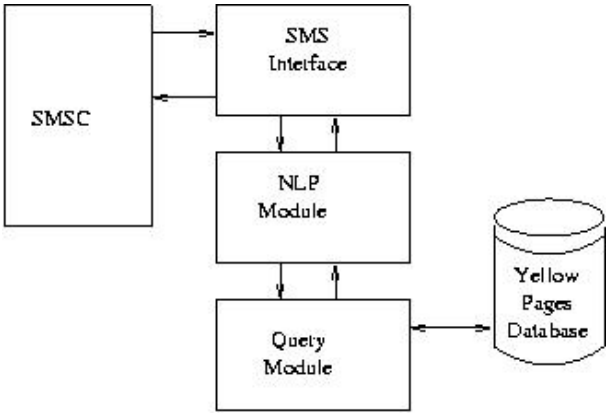

Figure 1: Overview of the system to query yellow pages on the mobile network.

The tagged information (T_1) is sent to the query module, which forms a SQL query to retrieve information from the database. In the event of no records being returned from the database, the NLP module generates another tagged list (T_2) which is used by the query module to search the database. This process is continued until one or more records are returned from the database to the query posed by the user. The generation of the tagged list (T_1, T_2, ...) depends on the initial query and is such that the SQL query has a higher probability of attracting records in the database.

A prototype of the system that can access the yellow pages directory for the Mumbai yellow pages directory was deployed for a major telecom operator in India. The next section describes the key advantage of using an NLP based strategy that allows the user the freedom of how and what they ask. The system answers intelligently (as a normal human would do) by giving answers when present in the database else giving approximate but close answers in the absence of an exact answer being available in the database.

3. ADVANTAGE OF USING NATURAL LANGUAGE INTERFACE

Designing he strategy of generating T_1, T_2, ... from the initial query of words is crucial and helps in digging out information from the database in the event of information being absent for the initial query words. This strategy gives our system an edge over using an ordinary keyword based search strategy. The strategy information is part of the NLP module (Figure 1). In addition the NLP module is supported by a taxonomy tree[1].

We establish the advantage of using such a strategy through actual examples[2].

Scenario - 1. *Suppose a user is looking for a* **Studio**

[1]A tree structure, that captures relationship between different words. Traversing through the tree one is able to relate different words. In literature it also goes by the name ontology tree.

[2]For Mumbai yellow pages directory

in Eastern Andheri[3]. *An ordinary search strategy would fetch: [1] EASTERN TRADERS, ANDH [2] EASTERN ELECTRONICS [3] M K EST, 73, M K EST, ANDH (E), 28590034, records from the yellow pages database while the NLP based strategy would enable extract appropriate and exact information, namely, [1] M K EST, 73, M K EST, ANDH (E), 28590034 [2] GEMINI STUDIOS, C/3, M.I.D.C., ST NO 11, ANDH (E), 28229933 [3] KAMAL AMROHI STUDIO, JOGESHWARI VIKHROLI LINK, ANDH (E), 28208026 [4] CHANDIVALI OUTDOOR STUDIO, CHANDIVALI RD, ANDH (E), 28521097*

While an ordinary search strategy would produce any results with any of the words in the query (studio, eastern Andheri) as the keywords, the NLP based strategy would be able to return better search results by understanding that the query intend to search for a studio in 'Eastern Andheri'.

Scenario - 2. *For a query* **Want to have Meduvada in Juhu**, *an keyword based search engine would not return any results while the NLP based system would return [1] UDIPI SHREE KRISHNA, JUHU CHURCH RD, JUHU, 26713178. Following may also be useful: [2] THE SEAFARER REST, LIONS, CNTRL JUHU BEACH, JUHU, 26162839 [3] SAPPHIRE, THE EMERALD, JUHU TARA RD, JUHU, 26611150 [4] SUBURBIA REST & BAR, GAYLAND HTL, JUHU TARA RD, JUHU, 26170999*

Scenario - 3. *Suppose a user queries* **looking for DTDC Couriers in Sanpada**. *In the absence of a DTDC courier in Sanpada the ordinary search strategy would produce no output, while a NLP based system would give close and appropriate records extracted from the yellow pages directory [1] DTDC STALLION ENTPS,STALLION ENTPS, A 31, VASHI PLAZA., SEC 17., VASHI*[4]*, 27894652 [2] DESK TO DESK COURIERS, 7, GR FLR, AMBASSY CENT, NARIMAN POINT, 56311357 [3] DESK TO DESK COURIERS, GALA NO 15, 1ST FLR, MEHTA STATE, NDHERI KURLA RD, ANDH(E), 56943478 [4] DTDC COURIERS, 15, MEHTA EST, AND-KURLA RD, ANDH(E), 56943477*

Scenario - 4. *For a query,* **Citibank ATM in Vashi**, *the keyword based search strategy returns no results (because of a Citibank ATM being absent in Vashi). On the other hand the NLP strategy based system is able to give information that is useful while suggesting that that there is no perfect fit for the query posed. It says No perfect fit for CITIBANK ATM IN VASHI. Hope this helps: [1] CITIBNK, PANCHEEL ARCD, SEC 5, AIROLI*[5]*, [2] INDUSIND BNK LTD, MANEK CPLX, SEC 29, VASHI, [3] UTI BNK SHP 1, PL 17, SHIV DARSHAN, SEC 4, VASHI [4] UTI BNK WARDHAMAN CHMBS, PL 84, SEC 17, VASHI, 27660066*

Scenario - 5. *For the query* **Breakfast in Taj** *while keyword based search would results in no record or any record having Taj as the company name, NLP based strategy results in correct results because of the ability to relate breakfast to a place which serves food. The records returned by NLP based strategy are [1] TAJ GROUP OF HTL, MANDLIK HSE, MANDLIK RD, COLABA, 22022626 [2] TAJ GROUP OF HTLS, MANDLIK RD, APOLLO BUNDER, COLABA, 56653366 [3] TAJ PRESIDENT, 90, G D SOMANI RD, CUFFE PARADE, 56650808 [4] TAJ LANDS END REGENT, LANDS END, BANDSTN, BDRA(W), 5668123*

Scenario - 6. *For a query* **Buying Jeans in Andheri** *ordinary search strategy produces no results while the NLP based strategy gives [1] IMAGE APPARELS P LTD, ARVIND CHMBS, WERN EXPRESS HIGHWAY, ANDH(E), 28224892 [2] LIVE IN JEANS, C-6, MIDC, RD NO 22, ANDH(E), 28252127 [3] APEX JEANS WEAR, 9/F, NANDJYOT INDL PREMISES, ANDH KURLA RD, ANDH(E), 28511891 [4] SINGPORE OLLECTION,DN RD, ANDH(W), 26209109*

Scenario - 7. *For the query* **Cable operator in Vashi**. *While the ordinary search strategy produces no results, the NLP based strategy gives [1] SSV CABLE P LTD,9, NR INDIAN BNK, LANDMARK CHS, SEC 14, VASHI, 27664073 [2] AASHISH CABLE NET INDIA P LTD,SEC 9 A, GURAV HALL, VASHI,(O)27655535. In addition it also list [3] SEVEN STAR CABLE, SHP-3, MINI JEWEL, OPP GTB BNK, SEVEN B'LOWS, ANDH(W), 26362675 and [4] UCN HATHWAY CABLE, 2ND FLR, STRAND CNMA BLDG, COLABA, 22812994 suggesting them as a possible alternative answers to the query.*

These examples clearly demonstrate the value add of using NLP based strategy to retrieve information from the database. In all the cases, demonstrated, the ordinary search strategy fails because of either the absence of the information in the yellow pages directory or the inability to extract more information from the query instead of treating all the query words as being key search words.

4. ALGORITHMS DRIVING THE NLP ENGINE

The proposed system, driven by the NLP engine[2] is able to provide *intelligent* answers to queries. For the purpose of developing the algorithms that drive the NLP engine raw queries received at the call center were analyzed to determine the type of queries. The information derived from a random set of 186 queries is given in Table 1. Essentially, the person querying for yellow pages information usually asked by the company name ('Microsoft') and one search word ('software'). While about 20% of the queries were based on the name of the company and its location.

The SMS query (a sequence of words) is the input to the NLP engine and the NLP engine generates a predetermined number of (word, tag) pairs. This (word, tag) pair is to be used by the `search_db_module` to query the database[6]. Initially, the SMS input the NLP engine is preprocessed as described in Algorithm 1.

[3]Andheri is a suburb in Mumbai

[4]Note that Sanpada and Vashi are adjacent suburbs in Mumbai

[5]Note that Airoli is one of the adjacent suburb of Vashi

[6]The search module makes an SQL query for the word in the tag column on the database

Total Queries analyzed	187
One Search Word and Company Name	91 (49 %)
Company Name and Location	37 (20 %)
Company name search only	20 (11 %)
One Search Word and Company Name and Location	14 (7 %)
One Search Word and Location	8 (4 %)
Two Search Words	5 (2.5 %)
One Search Word only	3 (1.6 %)
Unclassified	rest

Table 1: Statistics of actual Queries asked by users to Call Center Agents

The processing done by the system is based on the actual query. In fact the algorithms described in this section fall into four cases based on the content of the query. They are

1. Case I – Only `company_name` and `location` present in the query
2. Case II – Only `search_word` and `location` present in the query
3. Case III – Only `search_word` and `company_name` present in the query
4. Case IV – `search_word location` and `company_name` present in the query

The algorithms associated with each case are described in Algorithm 2, 3, 4, 5.

5. RESULTS

The database consists of more than one million businesses catering to all walks of life. Each business carries among other information the description of the business through a sequence of `search_word`[7], the `location` details of the business. The name of the business is the `company_name` tag. This database is searched by the search module using the (word, tag) pair generated by the NLP engine (see Algorithms 2, 3, 4, 5).

The system was piloted with a large telecom operator to test the performance of the system. Majority of the users who used the system were not associated with the development of the system. The queries posed by them were classified as being relevant questions or irrelevant questions[8] for the purpose of analysis. Example of irrelevant queries were essentially queries which were (a) Out of the yellow pages domain e.g. Cricket score, weather details, phone number of famous people, (b) Absurd queries e.g. AAAAA (sequence of meaningless words) or (c) Too General Queries e.g. India, Why?. Only the relevant questions were analyzed to determine the performance of the system. A total of 2000 queries were analyzed. Table 2 gives the distribution of relevant and irrelevant queries.

[7]a minimum of three and a maximum of twelve describe each business

[8]relevance to the yellow pages domain

Algorithm 1 Preprocessing

Given an SMS message $\mathcal{W}$ made of M words, namely w_1, w_2, $\cdots$, w_M
Given tags `company_name`, `search_word` and `location`
Given N number of search queries to be generated

for i = 1: M **do**
  Determine if w_i is a `search_word` (taxonomy tree) or `location` (location tree)
  if w_i is not `search_word` or `location` **then**
    Tag w_i as `company_name`
  end if
  Tagged $w_i \in$ { `company_name`, `search_word`, `location`}
end for

Every word in the SMS message $\mathcal{W}$ has been tagged; there are M words
for k = 1: M **do**
  if Tagged $w_k \in$ { `search_word`} **then**
    Using `tag:` information in taxonomy tree
    if w_k is not a `search_word` **then**
      Find $w'_k \in$ `search_word` traversing taxonomy tree width first
      $w_k \leftarrow w'_k$
    end if
  end if
end for

Let $w_c \in$ `company_name` and $w_l \in$ `location`, $w_s^1, w_s^2 \in$ `search_word`
Need to generate N search queries

Question Types	%
Relevant Questions	88.7
Irrelevant Questions	12.3

Table 2: Distribution of queries

To determine the performance of the system, a random set of 1000 users were themselves asked to rate the response of the system into one of the following three categories (satisfactory response by the system, not satisfactory and can not say). The performance of the system is shown in Table 3. The majority of the responses that the users though were incorrect was because they were aware of a business in their locality which did not show up as response to their queries. There are several reason for such a thing to happen, some of them are (a) the business details were not in the database, (b) more than one businesses satisfied the query and only the first three businesses were returned as the possible answers by the system.

Answer Types	%
Good Answers	86.0
Incorrect answers	12.7
Can't Say	1.3

Table 3: Analysis of the response

6. CONCLUSIONS

Algorithm 2 Case I: Only `company_name` and `location` present

if only $w_c \in$ `company_name` and $w_l \in$ `location` present **then**
  Find w'_c a company equivalent to w_c from the company table
  Find w'_l a location close to w_l from the location tree
  Find w''_c a company equivalent to w'_c from the company table
  Find w''_l a location close to w'_l from the location tree
  ...
  Q_1: Search w_c in `company_name` field and w_l in `location` field
  Q_2: Search w'_c in `company_name` field and w_l in `location` field
  Q_3: Search w_c in `company_name` field and w'_l in `location` field
  Q_4: Search w''_c in `company_name` field and w_l in `location` field
  Q_5: Search w_c in `company_name` field and w''_l in `location` field
  Q_6: Search w''_c in `company_name` field and w'_l in `location` field
  Q_7: Search w'_c in `company_name` field and w''_l in `location` field
  ...
end if

Algorithm 3 Case II: Only `search_word` and `location` present

if only $w_s^1, w_s^2 \in$ `search_word` and $w_l \in$ `location` present **then**
  Q_1: Search w_s^1 in `search_word`, w_s^2 in `search_word` field and w_l in `location` field

  Find w'^1_s close to w_s^1 from taxonomy tree
  Find w'^2_s close to w_s^2 from taxonomy tree
  Find w'_l a location close to w_l from the location tree
  ...
  Q_2: Search w'^1_s in `search_word`, w_s^2 in `search_word` field and w_l in `location` field
  Q_3: Search w_s^1 in `search_word`, w'^2_s in `search_word` field and w_l in `location` field
  Q_4: Search w_s^1 in `search_word`, w_s^2 in `search_word` field and w'_l in `location` field
  Q_5: Search w'^1_s in `search_word`, w_s^2 in `search_word` field and w'_l in `location` field
  Q_6: Search w_s^1 in `search_word`, w'^2_s in `search_word` field and w'_l in `location` field
  Q_7: Search w'^1_s in `search_word`, w'^2_s in `search_word` field and w'_l in `location` field
  ...
end if

Algorithm 4 Case III: Only `search_word` and `company_name` present

if only $w_s^1, w_s^2 \in$ `search_word` and $w_c \in$ `company_name` present **then**
  Q_1: Search w_s^1 in `search_word`, w_s^2 in `search_word` field and w_c in `company_name` field

  Find w'^1_s close to w_s^1 from taxonomy tree
  Find w'^2_s close to w_s^2 from taxonomy tree
  Find w'_c close to w_c from the company file
  ...
  Q_2: Search w'^1_s in `search_word`, w_s^2 in `search_word` field and w_c in `company_name`
  Q_3: Search w_s^1 in `search_word`, w'^2_s in `search_word` field and w_c in `company_name`
  Q_4: Search w_s^1 in `search_word`, w_s^2 in `search_word` field and w'_c in `company_name`
  Q_5: Search w'^1_s in `search_word`, w_s^2 in `search_word` field and w'_c in `company_name`
  Q_6: Search w_s^1 in `search_word`, w'^2_s in `search_word` field and w'_c in `company_name`
  Q_7: Search w'^1_s in `search_word`, w'^2_s in `search_word` field and w'_c in `company_name`
  ...
end if

Algorithm 5 Case IV: `search_word location` and `company_name` present

if $w_s^1, w_s^2 \in$ `search_word`, $w_c \in$ `company_name` and $w_l \in$ `location` **then**
  Q_1: Search w_s^1 , w_s^2 in `search_word`, w_c in `company_name`, w_l in `location`

  Find w'^1_s close to w_s^1 from taxonomy tree
  Find w'^2_s close to w_s^2 from taxonomy tree
  Find w'_c close to w_c from the company file
  Find w'_l a location close to w_l from the location file
  ...
  Q_2: Search w_s^1 , w_s^2 in `search_word` , w'_c in `company_name`, w_l in `location`
  Q_3: Search w_s^1 , w_s^2 in `search_word` , w_c in `company_name`, w'_l in `location`
  Q_4: Search w_s^1 , w_s^2 in `search_word` , w'_c in `company_name`, w'_l in `location`
  Q_5: Search w'^1_s , w_s^2 in `search_word` , w_c in `company_name`, w_l in `location`
  Q_6: Search w_s^1 , w'^2_s in `search_word` , w_c in `company_name`, w_l in `location`
  ...
  Q_α: Search w'^1_s , w'^2_s in `search_word` , w'_c in `company_name`, w'_l in `location`
  ...
  Q_N: Search $w''' \cdots '^1_s$, $w'' \cdots '^2_s$ in `search_word` , $w'' \cdots '_c$ in `company_name`, $w'' \cdots '_l$ in `location`
end if

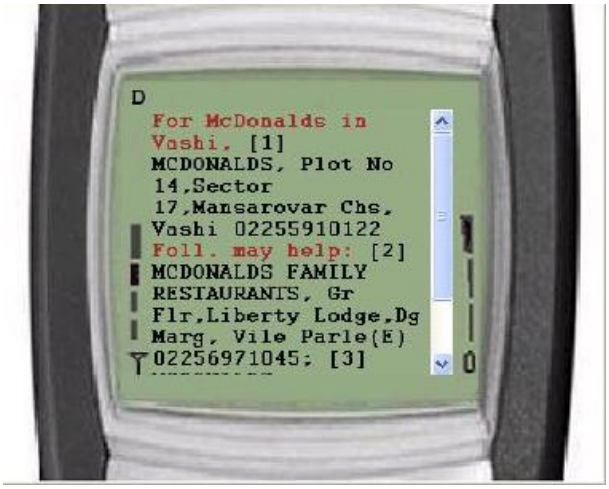

Figure 2: Real Example - Query with `company_name` and `location`.

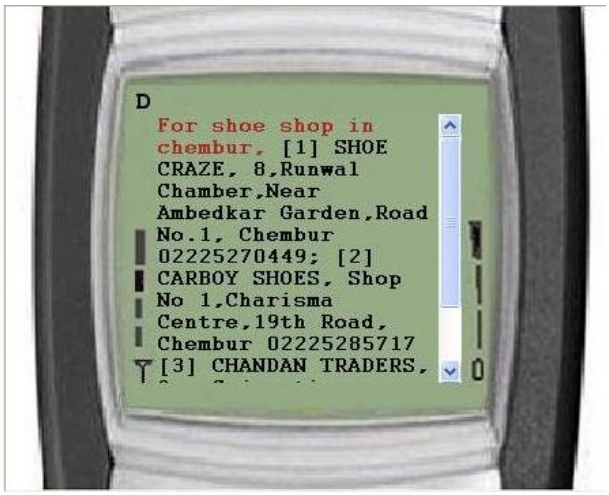

Figure 3: Real Example - Query with `search_word` and `location`.

Figure 4: Real Example - Query with `search_word` (spelled wrong!) and `location`.

In this paper we have developed a natural language mobile interface that gives the user an unconstrained mode of asking for information from the yellow pages directory, 24×7. The system is user friendly because it allows the user to access information in the yellow pages directory by posing the query in natural English, takes care of spelling mistakes and SMS lingo. The proposed interface is on a low cost SMS channel – the most popular and the cheapest mode of communication on the mobile network in developing countries. The system can be however easily configured to work on a WAP enabled phone where the user can be provided a text box to type in his query in natural English. The performance of the system piloted with a major telecom operator in India is very good. One of the salient features of the system is its ability to give the next best answer in the absence of exact information not being available in the database in addition to giving the user the flexibility of posing the query without constraining him to pose the query in a particular format. The natural language processing based engine is being proposed to query for classified advertisement information, music download and as a system to help rural folks in India get expert cultivation information among other value added mobile services.

7. REFERENCES

[1] JustDail. Talking Yellow Pages. In *http://www.justdial.com/*, 2004.

[2] S. Kopparapu, A. Srivastava, and P. Rao. Minimal parsing question answering system. In *International Conference on HCI*, 2007.

[3] PRWeb. Happy VAS customers translate directly into increased ARPU and profit. In *http://www.prweb.com/releases/2003/2/prweb58656.htm*, 2003.

[4] YellowLine. Infomedia Yellow Pages. In *http://www.yellowpages.co.in/*, 2004.