

Talking Agent* : An Interactive Natural Language Query Interface to Raw Information

- Rohit Kumar -

26. July. 2003

Some notes on the proposed Final Yr. Project from Speech Lab discussing the project and its various components and the initial ideas about its possible implementation.

* We can think of some name for this e.g. inquire, quest, jigyaasa, jaankari, jawaab

Basic Idea

To develop an interactive system that provides a natural language query based interface to raw textual information like news corpus, textbooks, help manuals and other information sources like websites. Further a speech interface in proposed to provide extra dimensions of interaction.

The block diagram of the system can be as follows.

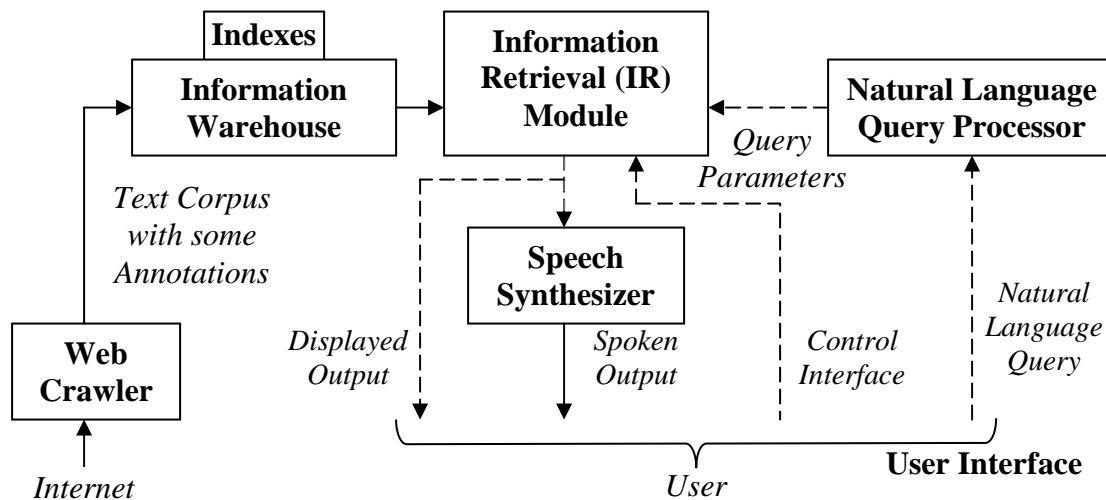


Figure 1. Block Diagram of Talking Agent* System

Each of these blocks, their requirements, design considerations and possible implementations are discussed further.

Web Crawler

The Web Crawler will be website specific with some general modules. It will basically be a HTML parser which will get the relevant text from websites. Further the web crawler should crawl i.e. it should be able to span web pages by following the links given a starting point and some parameters like depth, domain name restrictions etc. The

Web Crawler would save the relevant text and also a log of information about from which place the information has been collected at what time and date and what were the last updates etc.

The text corpus which is to be used for information extraction is required to be having proper annotation marking elements like dates, places, titles, names, etc. (other relevant elements we might find suitable)

- Also the information like date, author, topic, should be embedded
- The annotation could be manual or automatic or by a combination of the two using text processing modules with minimal NLP
- The text corpus should be in proper notation when we are working with Indian Languages like UNICODE and proper support modules should be developed for the notation

Information Warehouse

The annotated text corpus is to be indexed suitably and stored in the information warehouse. The indexing and storage should be aimed at facilitating the information retrieval (IR) module. The following are desirable:

- Rapid Searching
 - Maximum processing should be done offline while warehousing
- Indexes relevant to the possible kinds of query
- Multiple Indexes e.g. by -
 - Date
 - Keywords (dictionary based or by NLP to identify keywords like proper nouns etc.)
 - Context
- Data may be stored in levels of detail: headline, paragraph, discussion, lecture, etc
- Also some clustering can be done as in www.AllTheWeb.com
- If we are working for new sites like BBC we can use also see links and even pictures

There isn't much to be done with Indexing Techniques. Common data structures like graphs and hash tables may be used optimally.

The basic design of indexes / storage techniques will be dictated by the requirements of IR module. We can use parameters like:

- Count of occurrence of keywords in a particular place
- Position of occurrence of keywords

Also page ranking (as in Google) can be used for internal links among pages on a particular website.

References:

- IE/IR papers in ICON 2003: Document Space Model and Phrase Grammar
- Text summarization and categorization work to find out what all other features are relevant and have been used and how to extract them

Information Retrieval (IR) Module

The IR module is a basic search engine effectively. It is supposed to find the most relevant information from the warehouse given certain query with various kinds of parameters.

The IR module should select the top few most relevant articles and rank them. Also it should append and prepend the extract text with proper messages.

IR modules should be simply using the best of various indexes available with us and should take us to the best results very fast. The indexed stored while warehousing should take care of maximum processing offline.

Saving Space should not be at the cost of Speed.

- The IR modules and the query processing modules should have a socket interface where they get the query packets and give out the packets of extracted information
- Also some session management should be provided like <Next> ... <Prev> etc.
- Advanced search facilities like search within search results

IR module and warehouse are highly depended on the kind of queries we want to process i.e. what kind of parameters. IR module will be relatively kept very simple in the way that is should simply look at some query parameters and accordingly access the indexes in the ware house to calculate the rank of relevant documents based on certain parameters like date, keywords, context, etc.

Natural Language Query Processing Module

The NLQP Module is the crux of the design of the whole system. The basic purpose of the NLQP module is to accept a Query in Natural Language from the user and extract query parameters for the Query as well the context of the on going interaction with a particular user and also in some more contexts like current date, some hot topic, etc. which may be tuned from time to time by the webmaster (see User Interface).

Simply speaking a NLQP module is a supposed to give some keywords on the basis of which the IR modules can lookup the indexes and find the most relevant answers. We need to come up with ways to find these keywords from Queries.

One way is creating a knowledge base (if we can call it that) of the various starting questions that a person can ask. Besides we can keep some generic ones which can fit in all exceptional situations. Then for these questions we say that the words

appearing in these particular places will be keywords and the context extra can be manually written. In a way this is similar to writing the templates of AIML. Patterns will be similar only the response will be in form of Query parameters. So the methods to find the most suitable pattern for a Query can be taken from AIML Interpreter.

This can only be done for the first question or may be the second too but beyond that it gets too specific. And we can then use simple keyword detection by finding proper nouns and some more important things like kyaa, kab, kisme, kahan, i.e. query classifiers.

Another approach can be to build a strong NLP module to generate a parse tree of the query and then use some rules to find that this is most relevant and things like that. We need to do some explorations into the way this has been done.

User Interface

The user interface should be as separated from the NLQP and IR modules as shown in Figure I. There should be Sockets based interface (shown by dotted lines) so that the system can be at on a different machine and users can access from somewhere else.

Also several things can be thought of about how to make the interface more interactive. Basically the interface would allow typing query in natural language. This can be made good when we are working with Indian Languages by providing an on screen keyboard and some dictionary based support to suggest words.

On the output end, the texts will be presented in displayed as well as spoken format. Some sort of synchronization in the two will improve the accuracy a lot. We can come up with our own Wav file format with the transcribed text embedded within the file itself. Also a control interface to go to previous, next, related, suggested links should be possible. Further display of related media like images (if any) can be support in the long run.

Speech Synthesis Module

The output to the user will be Interactive and speech will be one of the modalities. We already have the engine for this and this wont take a great time to be done away with once we start getting good results. Only we will need to write some good converters and some basic Text Processing modules.

Possible plan of action

1. Explore the work that has been done related to the NLQP module and for what kind of parameters have been used for Text Summarization so that we can use it Indexes
2. Start collecting text corpuses and come up with a Crawler first

3. Design the NLQP module because it designs the rest of the system. Think of all the kinds of Query we might be processing
4. Now start working parallelly from both direction on NLQP and on Warehousing / Retrieval.
5. Lot of experiments and we will come up with something
6. Then we can see how to proceed

Some issues

Thinking about them