Information-Demanding Question Answering System

Calkin A.S. MONTERO and Kenji ARAKI
Graduate School of Information Science and Technology, Hokkaido University,
Kita 14-jo Nishi 9-chome, Kita-ku, Sapporo, 060-0814 Japan
Tel. & Fax:+81-11-706-7389
E-mail:{calkin,araki}@media.eng.hokudai.ac.jp

Abstract— **Natural language question answering (QA) systems that aim to find an accurate answer to a given question have the power to revolutionize computer applications. This paper presents an approach to QA system in which the application not only aims to find a suitable answer to a question but also tries to bring the user-computer interaction closer to human-human interaction through a casual conversation or "chat" with the user. Experiments results are presented showing how human-like computer behavior could improve a QA system.**

I. Introduction

QA system has become a powerful paradigm in Artificial Intelligence (AI) extending beyond AI systems to query processing in database systems and to many analytical tasks that involve information gathering, correlation and analysis. In open domain QA systems, the user can ask any kind of question since there is not restriction on the scope of the questions. Hence, most open domain QA systems use large text collection from which they attempt to extract relevant answers. The rapidly increasing availability of information on the World Wide Web (WWW) has made the Web an attractive resource [1], [2]. This availability has made QA system a compelling framework for finding information that closely matches user's needs by providing answers instead of retrieving documents.

Several approaches to open domain QA system have been proposed. Prager et al. [3] introduced an indexing approach using predictive annotation technique, a methodology for indexing texts for fact-seeking QA systems. Pattern-based QA system approach has been shown to perform satisfactorily [4]. However, those approaches focus on finding possible answers to a user question using just the information that can be extracted from the question itself. In many cases, this task is very hard since the question does not contain sufficient information to find a suitable answer. For example, the incomplete question "Who is a diamond producer?" has a wide spectrum of possible answers when the knowledge base is the Web. Hence, the user's needs are hard to match.

To limit the spectrum of possible suitable answers, we propose an "information-demanding" QA system that acquires more precise information from the user regarding the question by means of *chatting* with the user. This approach also aims to smooth the user-computer interaction. In fact, human-computer conversation (HCC), which is part of natural language processing technology and is among the oldest, most important, and most current areas of Artificial Intelligence (AI), has reached a similar stage of development as some better-known areas of language processing, like Information Extraction (IE) and Machine Translation (MT). One of the most famous examples of HCC is ELIZA [5], a computer program that interviews a psychological patient without limiting words. SHRDLU [6] is another well-known dialogue program that carries on a simple dialogue (via teletype) with a user about a small world of objects.

Recently the development of dialogue systems has increased exponentially with advances in areas like dialogue management and context tracking, so that we have systems like JUPITER [7] capable of solving a domain-limited task while interacting with the user.

By applying HCC techniques to open domain QA system, we aim to simplify the possible answer selection task and, at the same time, to smooth the user-computer interaction.

In the following paper an overview of the system is provided in Section II, followed by a detailed explanation of each process in Section III. The performed experiments and obtained results are described in Section IV, and a discussion of the obtained results is stated in Section V. Finally a conclusion with reflections on future directions for the system is given in Section VI.

II. Fundamental Concept

We aim to achieve a QA system capable of holding a human-like interaction with the user. Fig. 1 shows our system overview. As shown, the QA system is based on information-demanding, by means of *chatting* with the user (see III.A).

The system processes the user utterances using a morphological analyzer and sends a formed query to the Web search engine (using Google API [8]). The retrieved documents are processed for extraction of possible answers to the user's request. A more detailed explanation of each process is described hereunder.

III. Information-Demanding QA System

A. Dialogue Management

To build an information-demanding QA system, we created a 'chat-bot-like ELIZA-clone' oriented to demand and acquire information from the user. A chat-bot is a computer program

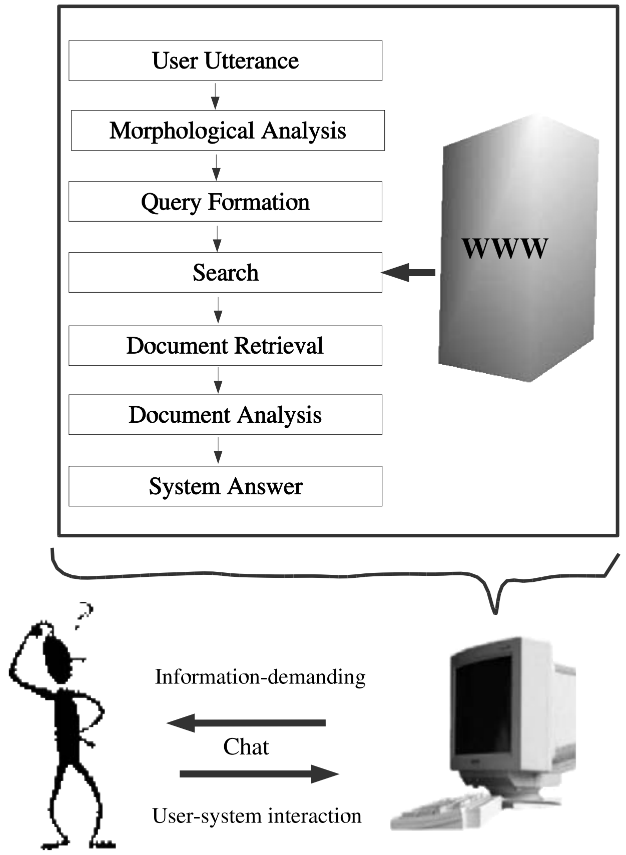

Fig. 1. Information-demanding QA: System Overview

that simulates human chatting. Chat-bots are found widely on the Web and are typically used in applications for e-commerce customer service and the like, as the chatting is limited to specialized areas. With the use of an ELIZA-like chat-bot oriented to obtain concrete information from the user about his/her request, we can better satisfy the user's needs and at the same time achieve a computer interaction more like human chatting.

It is the acquisition of information directly from the user - information that is in addition to the question itself - that is one of the advantages an information-demanding QA system could have over a typical QA system, which depends solely on the information extracted from the user's question. This is particularly useful in cases where the user's question lacks sufficient information to elicit a suitable answer, as is the case of many "Who is [noun][1]?" or "Who are [noun]?" questions.

The following chat example illustrates one of these cases:

.....
System: Hi! How can I help you?
User: hi
System: How do you do? Please tell me your question
User: yeah...
System: Pardon me?
User: who is a diamond producer?
System: I'll search the Web. It may take a while and the possible answers may be many. It'll help if you give me more information. What exactly are you looking for?
User: I m looking for a diamond producer in America
System: So you said "diamond producer in America"?
.....

The chat is managed using scripts to simulate 'the system understanding' of the user's utterances. The question "who is a diamond producer?" when given to a typical QA system will retrieve about 340,000 Web pages, with information on diamond producers throughout the world, and since it is a "Who" question, most QA systems will search for a "person's name" as a possible answer.

This overabundance of information makes it very difficult for the system to find the most suitable answer, regardless of how good its question analysis is. However, with an information-demanding QA system more specific information about what the user is trying to find could be obtained. In the example shown above, the acquisition of the new data " America", (see III.B) reduced the number of retrieved pages by

[1] [ ] encloses here a part of speech

TABLE I
WEB-QUESTION ANALYSIS

Web-question detected
"who is a diamond producer?"
Morphological analysis

Words of the question	Part of speech	Original form
who	WP(wh-pronoun)	who
is	VBZ(verb to be, 3rd person sing. present)	be
a	DT(determiner)	a
diamond	NN(noun singular)	diamond
producer	NN(noun singular)	producer
?	SENT(end of sentence)	?

Formed query
diamond producer

approximately 65% and provided information about 'Canada' as diamond producer in *North* America on the first few Web pages. Hence, information-demanding helps to greatly reduce the number of Web pages to be processed for possible answers.

It is worth mentioning that another advantage of this QA system is seen during question analysis: The question classification task - performed by most QA systems for narrowing the spectrum of possible answers [9], [10] - was avoided because the user provides the system with all needed information for finding possible sought-after answers.

In the case that the user does not ask any *Web question*[2] the system performs trivial chatting with the user. The system counts on 174 scripts, including idioms, colloquialisms, etc., in the form of a condition-action rules with a pattern on the left-hand side (LHS) and its associated set of responses on the right-hand side (RHS), for dealing with the user chat. If no script is found for answering the user's utterance, then the system performs an ELIZA-like behavior. The following chat illustrates the system trivial behavior:

.....
System: Hi! If you have a question I'll search on Internet.
User: hello, my name is Joe
System: hello Joe, nice to meet you.
User: hi
System: Hi! I'll search on Internet for you.
User: thanks
System: Uhmm, please continue. (ELIZA-like behavior)
User: what is your name?
System: I am a computer program. My name is Dana.
.....
User: I'm not asking anything
System: Let's discuss further about that. (ELIZA-like behavior)
User: i have to go
System: I hope to talk to you again.
.....

In the case that the asked question does not require additional information in order to be suitably answered or that the user does not want to provide the system with additional information, the user can use fixed expressions to instruct the system to use only the information extracted from the Web-question.

B. Query Formation

The system determines a Web-question by detecting question words (what, who, when, which, etc; with the exception of some questions about the system itself -"what is your name?", "how are you?" and the like-). When the system is given a Web-question, it performs a morphological analysis tagging the question. This process is needed in order to extract specific part of speech from the question for building a query. To perform this task, a publicly available tool called TreeTagger [11] is used. This tool is a part of speech tagger based on a decision tree. It has been trained over a large corpus of English, to assign the most likely sense to each word it parses. This tool is accessed dynamically from our system and it has shown to tag correctly most of the input cases.

The focus of this research is on factual questions. More specifically, since 'who' questions tend to lack information (as shown in the first chat example in the previous section), scripts were specifically designed to deal with them. The query is formed by extracting nouns, adjectives, adverbs and verbs (with some exception, like the verb "to be") from the given question. Table I shows the analysis of the previous example.

While the question is being analyzed the system attempts to obtain information from the user by chatting. After the Web-question is detected the following user utterance is morphologically analyzed for extraction of valuable information -that is nouns, adjectives, etc.- to be added to the previously formed query. In our example, the new information obtained from the user is "America"(NN); thus, the new query becomes *"diamond producer America"*. This "augmented query" is then sent to the search engine and documents containing possible answers are retrieved.

It is worth mentioning that the augmented query does not have to have necessarily direct relation with the sought after answer. To be specific, in the previous example, the augmented query is *"diamond producer America"*, although the sought after answer is "Canada". Therefore, the extraction and acquisition of *appropriate keywords* (from the question and the user) is enough to retrieve documents that potentially comprise the sought after answer. Thus, nonetheless the augmented query of our example does not include the whole noun "*North* America"

[2] A question whose answer is to be found searching the Web

within it, the added new information -America- is enough to find documents that contain information about "...Canada ...diamond producer...*North* America".

C. Document Retrieval and Answer Selection

Since the documents retrieved form the Web are automatically ranked by the search engine on their relevance to the query and since the query is formed by demanding precise information from the user, we believe that the possible sought-after answers could be found within the first few retrieved documents. Therefore, the system analyzes only the first 20 HTML Web pages from the thousands retrieved. The system parses the HTML Web pages using HTML-Parser [12] and segments each document into sentences using LingPipe [13]. From those sentences, the ones which contain keywords from the query are extracted and ranked according of the number of keywords from the query that they submit using (1).

$$KeywordsinaSentence(KW_S) = \frac{n+1}{2} \qquad (1)$$

where KW_S is a threshold and 'n' is the number of keywords in the query. KW_S is set to be more than a half of the total of words in the query. Thus, sentences comprising KW_S or higher number of words from the query are considered to be potential possible answers.

Using the previous question-example, possible answers extracted from the documents retrieved (with the query: 'diamond producer America' –> n = 3; $KW_S = 2$) are:

.....

–*Canada*: World's Third Largest **Diamond Producer**, Diamonds Net (Rapaport, January 4, 2004) According to a research paper released by Statistics...–

–*Canada*'s diamond industry third-largest in world: Statistics *Canada*, OTTAWA-In just five years, *Canada*'s burgeoning **diamond** business has put the country on track to become the third-largest **producer** in the world, Statistics *Canada*...–

.....

In the case that none of the sentences contains a number of keywords $\geq KW_S$, then a new threshold is set by reducing KW_S by 1.

IV. Experiments and Results

One of the most notable difference between our information-demanding QA system and a typical QA system (besides the smooth interaction with the user) is that the agent providing the needed missing information is the user; thus, the question does not need to be rigidly classified. Therefore, as was stated before, the question classification task could be avoided. To evaluate the effectiveness of our information-demanding QA system, we compare its performance to that of a typical QA system and to the performance of AnswerBus QA system (publicly available on the Web [14]). The typical QA system stated before was separately built. We describe concisely its algorithm hereunder.

A. A Typical QA System

A typical QA system performs several tasks that lead up to the 'user's question understanding' and therefore lead to selection of the best suitable answer to the user request. Four of those tasks are given due to their high level of importance: (a) question classification task, (b) query formation task, (c) document retrieval task (from the system knowledge database) and (d) answer selection task. Previous research [9], [10] has focused on the question classification task. The question classification task is important when selecting an answer due to the ability to narrow the spectrum of possible answer candidates. The typical QA system used during the experiment uses probabilistic question classification to classify the questions [15].

Question classification was defined as the task whereby given a question, the cluster in which that question is more probable to appear is selected from *n clusters*. Those *n clusters* represent *n categories*. We assumed 24 clusters: ABBREVIATION, ANIMAL, ART, BODY, COLOR, COUNTRY, CURRENCY, DATE, DEFINITION, DESCRIPTION, GROUPS, EXPANSION, FOOD, ENTITY, GEN.PLACE, MANNER, MEDICINE, PRODUCT, PERSON, PERCENT, REASON, SUBSTANCE, SYNONYMOUS, TRANSPORTATION. A first and second order Markov Model were built for each, and the Markov Models were combined using a linear combination. Since the Markov Model suffers from sparseness, "valuable features" were extracted from each cluster. Those features are 'named entities', 'nouns' and 'adjectives'.

To deal with the problem of unseen or unknown words that may appear in the test data, a combination of Backoff with Good-Turing smoothing technique [16] was used. As training data, 3,865 questions from a corpus publicly available [17] were selected and distributed into the 24 clusters in order to build Markov Models for them. As test data, 250 TREC10[3] questions were distributed into 24 sets according to each cluster. Despite its simplicity, this system could achieve an accuracy of 81.3% classifying individual questions and 21 out of 24 of the test data sets where correctly classified according to its cluster or category. This means 91.6% of accuracy for classification of the sets, at the same time avoiding computationally expensive semantic, sintactical and linguistic analyses.

Once the question is classified, a morphological analysis is performed and a query is formed extracting keywords from the question. From this query, nouns and verbs are expanded with their synonyms using WordNet[18]. As of 2003 the WordNet database contains about 140,000 words organized in over 110,000 synsets for a total of 195,000 word-sense pair. Every synset contains a group of synonymous words; words typically participate in several synsets. The synonyms obtained from WordNet are used to create different queries in order to be sent to the search engine and retrieve documents from the Web.

As stated before, the classification of a question is important

[3] 10th Text REtrieval Conference

TABLE II
COMPARISON BETWEEN TYPICAL QA SYSTEM AND OUR PROPOSED QA SYSTEM

System	Highly Related	Related	Barely Related	No Extraction
Typical QA system	12.0%	45.5%	32.0%	10.5%
Proposed QA system	25.0%	56.5%	18.5%	-

TABLE III
COMPARISON BETWEEN ANSWERBUS AND OUR PROPOSED QA SYSTEM

System	Highly Related	Related	Barely Related	No Extraction
AnswerBus	30.0%	20.0%	37.5%	12.5%
Proposed QA system	25.0%	56.5%	18.5%	-

for a typical QA system since according to each category of questions, set of answer patterns are built in order to be used for extracting possible answers from the documents retrieved.

For example, some answer patterns for the category ABBREVIATION of this typical QA system are:

$(@_1)$?/[possible answer]/NN/$(@_2)$?/.
$(@_1)$?/NN/$(@_2)$?/NP*/$(@_3)$?/abbreviated/[possible answer]/.
$(@_1)$?/NP*/$(@_2)$?/NN/$(@_3)$?/acronymous/[possible answer]/.
$(@_1)$?/NN/stands for/[possible answer]/.

where $@_n$ represents possible text, NP and NN are proper nouns and other nouns from the question. For the question “what does NASA stand for?”, which was classified correctly as ABBREVIATION, sentences extracted as possible answers were:

.....
–NASA stands “for the benefit of all”.–
–In the United States, NASA stands for the National Aeronautics and Space Administration.–
.....

B. Results

As stated above, this experiment tried to evaluate the effectiveness of our information-demanding QA system. The experiment compared how well the possible answers extracted by the system, the possible answers extracted by the typical QA system and the possible answers extracted by AnswerBus related to the user sought-after answer. The sentences extracted as possible answers were evaluated according to their number of keywords *(kw)* as Highly Related ($kw > KW_S$), Related ($kw = KW_S$) or Barely Related ($kw < KW_S$) to the user sought-after answer. The possibility of No Extraction was contemplated as well. We selected 40 questions from corpora publicly available [14], [17]. Results are shown in Table II and Table III.

The results show that the information-demanding QA system performs better than the other systems (around 81% of the sentences extracted as possible answers were related to the user's sought-after response).

V. DISCUSSION

The experiments showed that the implementation of an information-demanding QA system affords more accurate possible answer extraction. However, despite the huge amount of information available on the WWW, there were still cases in which the none of the three systems performed well. For example the question “who was the medieval classic hero that later became the king of Denmark?” (from [14]) had No Extraction, using the typical QA system and AnswerBus, and had the following sentences as possible answers using the information-demanding QA system:

.....
–His (putative; Harald never recognized him) son Sweyn Forkbeard *became King* of *Denmark*, Norway and England.–
–Arthur, called the first ‘worthy’ of the Middle Ages, the British Charlemagne, famous in history, legend, and romance, *became* a renowned *king* in British History around whom an epic literature grew up over time, who, himself, evolved in *medieval* romance into the central figure of numerous tales about his knights, many of whom *became* celebrated figures themselves.–
.....

These answers are barely related to the user sought-after answer. It can be seen from this example that even though a suitable answer could not always be given to the user, this information-demanding QA system, despite its very simple mechanism, always attempts to find sentences that could match the user's needs. Thus, it can considered a promising approach.

VI. CONCLUSION

In order to obtain a betterment matching user's needs as well as to achieve a smoother user-computer interaction, this paper proposed a simple information-demanding QA system, focusing on “who” questions, that uses a basic ELIZA-like chat bot. Scripts were used as a means of dialogue management, simulating the system understanding of the user's utterance. The information obtained from the user is useful for improving the query during the query formation process. As a result, the number of Web pages retrieved was greatly reduced and possible answers appeared within the first few Web pages, limiting the number of documents requiring processing. Preliminary experiments showed that with an information-demanding system, the user's sought-after answer can be more precisely extracted. Future works will be oriented toward broadening the chat-bot semantic database in order to deal with a wider spectrum of questions and with more user colloquialisms.

References

[1] C. Kowk, O. Etzioni, D. Weld, “Scaling Question Answering to the Web,” in Proceedings of the 10th International WWW Conference *(WWW10)*, 2001, pp. 150-161.

[2] S. Chakrabarti, M. van der Berg, B. Dom, “Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery,” in Proceedings of the 8th International WWW Conference *(WWW8)*, 1999. Also in Computer Networks, Vol. 31, No. 11-16, 1999, pp. 1623-1640.

[3] J. M. Prager, D. R. Radev, E. W. Brown, A. R. Coden, V. Samn, “The Use of Predictive Annotation for Question-Answering in TREC8,” in Proceedings of the 8th Text REtrieval Conference *(TREC8)*, National Institute of Standards and Technology (NIST) Special Publication 500-246, 1999, pp 399-411.

[4] M. M. Soubbotin, S. M. Soubbotin, “Patterns of Potential Answer Expressions as Clue to the Right Answer,” in Proceedings of the 10th Text REtrieval Conference *(TREC10)*, NIST Special Publication, 2001, pp. 175-182.

[5] J. Weizenbaum, “ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine,” Communications of the ACM 9, No.1, 1966, pp. 36-45.

[6] T. Winograd, “Procedures as a Representation for Data in a Computer Program for Understanding Natural Language,” Massachusetts Institute of Technology (MIT) AI Technical Report 23, February 1971.

[7] V. Zue et al., “JUPITER: A Telephone-Based Conversational Interface for Weather Information,” IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 1, January 2000, pp. 85-96.

[8] Google API for Perl. Google Web APIs (beta) (2003).
http://www.google.com/apis/

[9] X. Li, D. Roth, “Learning Question Classifier,” in Proceedings of the 19th International Conference on Computational Linguistics (COLING’02), 2002, pp. 556-562.

[10] D. Zhang, W. Lee, “Question Classification Using Support Vector Machine,” in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 26-32.

[11] H. Schmid, “Probabilistic Part-of-Speech Tagging Using Decision Trees,” in Proceedings of the First International Conference on New Methods in Natural Language Processing (NemLap-94), Manchester, England, 1994, pp. 44-49.

[12] G. Aas, HTML::Parser.(2003)
http://search.cpan.org/ gaas/HTML-Parser-3.35/Parser.pm

[13] Alias-i,Inc. Alias-i LingPipe. (2003)
http://www.alias-i.com/lingpipe/

[14] Zheng Z.: AnswerBus Question Corpus Database. (2003)
http://134.96.68.36/corpus/answerbus.shtml

[15] C. Montero, K. Araki, “Probabilistic Question Classification,” in Proceedings of the IEICE General Conference, Japan, 2004, pp. 49,

[16] W. Li, “Question Classification Using Language Modeling,” Center of Intelligent Information Retrieval (CIIR). Technical Report, 2002.
http://ciir.cs.umass.edu/pubfiles/ir-259.pdf

[17] Cognitive Computation Group at University of Illinois.
http://l2r.cs.uiuc.edu/˜cogcomp/

[18] WorNet: A Lexical Database for the English Language.
Cognitive Science Laboratory, Princeton University.
http://www.cogsci.princeton.edu/˜wn/