DERIVING DISAMBIGUOUS QUERIES IN A SPOKEN INTERACTIVE ODQA SYSTEM

Chiori Hori, Takaaki Hori, Hideki Isozaki,
Eisaku Maeda and Shigeru Katagiri

NTT Communication Science Laboratories
Nippon Telegraph and Telephone Corporation
{chiori,hori,isozaki}@cslab.kecl.ntt.co.jp

Sadaoki Furui

Department of Computer Science
Tokyo Institute of Technology
furui@cs.titech.ac.jp

ABSTRACT

Recently, Open-Domain Question Answering (ODQA) systems that can extract an exact answer from large text corpora based on text input are intensively being investigated. However, information in the first question input by a user is not usually enough to yield the answer desired. Interactions for collecting additional information to accomplish QA is needed. In order to construct more exact and convenient ODQA systems, this paper proposes an interaction approach for spoken interactive ODQA systems. With low reliabilities of answer hypotheses obtained by an ODQA system, the system automatically derives Disambiguous Queries (DQs) that draw out additional information. The additional information based on the DQs should contribute to distinguishing effectively an exact answer and supplementing lacking information by recognition errors. In our spoken interactive ODQA system, **SPIQA**, spoken questions are recognized by an ASR system and DQs are automatically generated to disambiguate the transcribed questions. The appropriateness of the derived DQs has been confirmed by comparison with manually determined ones.

1. INTRODUCTION

In the Spoken Language Processing (SLP) field, human and machine dialogue systems using speech interface have been intensively researched, some of which are marketed in phone systems, i.e. air ticket reservations, information retrieval for shops or stock prices. Such conversational dialogues that exchange information through Question Answering (QA) are a natural communication modality. However, state-of-the-art dialogue systems only operate for Specific-Domain Question Answering (SDQA). In order to realize more natural communication between human and machine, spoken dialogue systems for open domains are necessary. Especially Open-Domain Question Answering (ODQA) is an important function in natural communication. Our goal is to construct a spoken interactive ODQA system, which includes an ASR system and an ODQA system. In order to clarify the problems in accomplishing the spoken interactive ODQA systems, QA systems are classified into a number of groups depending on their target domains, interfaces and interactions that bring out additional information from users to accomplish tasks in Table 1. In this table, text and speech denotes text input and speech input, respectively. The term "*addition*" represents additional information queried by the QA systems. This additional information is other than the information in the user's first questions.

Table 1. Dialogue domain and data structure for QA systems

target domain		specific	open
data structure		knowledge DB	unstructured text
text	without *addition*	CHAT-80 [2]	FALCON [3]
	with *addition*	MYCIN [4]	(**SPIQA**∗)
speech	without *addition*	Harpy [5]	VAQA [7]
	with *addition*	JUPITER [6]	(**SPIQA**∗)

∗ **SPIQA** is our proposed system.

Recently, ODQA that extract answers from large text corpora, such as newspaper texts, has been intensively investigated in the natural language processing (NLP) field. The Text REtrieval Conference (TREC), co-sponsored by NIST and DARPA, has had an ODQA track since 1999 (TREC-8) [1]. Although the ODQA task is one of Information Retrieval (IR) issues, the ODQA systems return an actual answer rather than a ranked list of documents in response to a question written in natural language. On the other hand SDQA has been researched in the Artificial Intelligence (AI) field. The difference between SDQA systems and ODQA systems is in their data structure. Since information in a specific domain can be arranged in a table, the SDQA systems such as CHAT-80 [2] can be accomplished QA by table lookup techniques. On the contrary, since information in an open domain is scattered in large unstructured text corpora, the table-look-up technique cannot be applied.

Hypothetically, ODQA systems could be built from combining SDQA systems which include information tables for all different topics. This quasi-ODQA system might be able to answer user's question by switching to SDQA systems depending on topics of user's questions. However, it is very difficult to represent all information in unstructured text corpora using tables. The current ODQA system for large newspaper text and broadcast news transcription such as FALCON [3] extract answers by matching user's intention in questions to the answer classes. In these systems, supposing that the user's intention is a person's name, the ODQA system extracts some of person names in the retrieved paragraphs/documents which correspond to keywords in the user's question.

In order to obtain more exact answers to questions, some QA systems have interactions with users that can capture additional information to accomplish tasks. The QA systems with such interactions are denoted interactive QA systems. For instance, the expert system MYCIN [4] is an interactive SDQA system that diagnoses certain infectious diseases through a text dialogue. In this system, all solutions for the diagnosis have been designed in dialogue sce-

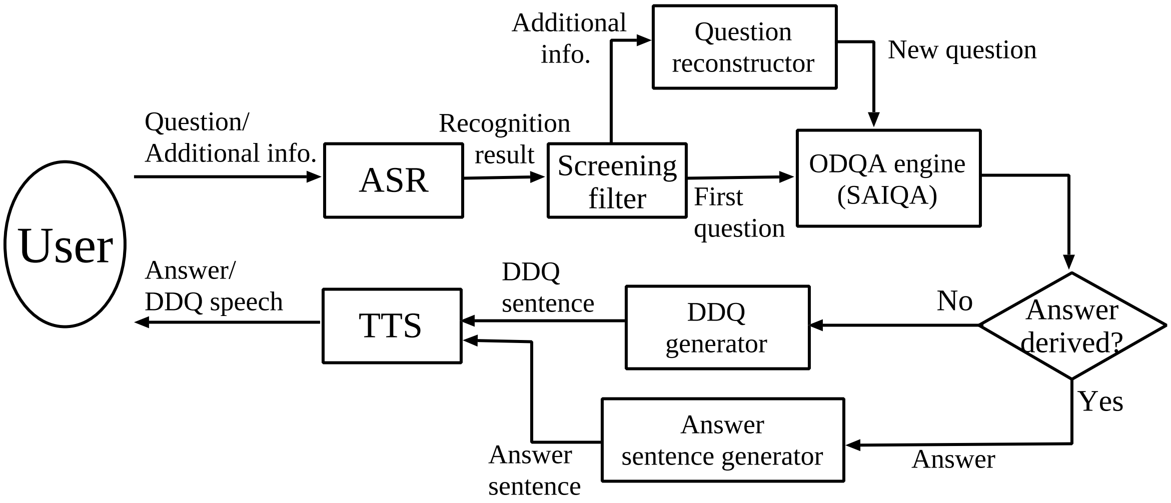

Fig. 1. Components and data flow in SPIQA.

narios using a knowledge database and IF-THEN rules. However, designing queries for additional information in an open domain for constructing interactive ODQA systems is not straightforward.

Since QA systems with speech interaction can be expected to exchange information more naturally, some spoken QA systems have been investigated. For instance, Harpy [5] is an SDQA system for academic journal paper retrieval which does not query additional information. Recently, an interactive SDQA system for worldwide weather forecast information retrieval over the telephone using spoken dialogue, JUPITER [6], was constructed. In such spoken QA systems, recognition errors should be an important consideration in system design. Recently, the spoken ODQA system, Voice-Activated Question Answering (VAQA) system [7] has been constructed. This system includes the ODQA system, FALCON [3] using speech interface instead of text input. The transcribed question is confirmed by the user. However, VAQA doesn't query for additional information other than the first questions.

In order to construct spoken interactive ODQA systems, we have to address some issues:

1. ODQA's problem:
   Answers are not in a table and scattered in unstructured text.
2. Interactive ODQA's problem:
   Since user's questions are not restricted, system queries for additional information to extract answers and effective interaction strategies using such queries cannot be prepared before user's question input.
3. Spoken QA's problem:
   Recognition errors degrade the performance of QA systems. Some indispensable information for answer extraction are deleted or substituted by other words.

In this paper, we'd like to propose an interaction approach based on disambiguation of users' questions for spoken interactive ODQA systems. In addition, our Spoken Interactive ODQA system, **SPIQA**, which includes an ASR system and an ODQA system is introduced.

2. SPOKEN INTERACTIVE QA SYSTEM: SPIQA

Figure 1 shows the components of our spoken interactive QA system, SPIQA and the flow of data through the system. This system includes an ASR system [8], a screening filter using a summarization method [9], an ODQA engine (**SAIQA**) [10] for a Japanese newspaper text corpus, and a **DDQ** (Deriving Disambiguous Queries) module.

ASR system

Our ASR system is based on the WFST approach that offers a unified framework representing various knowledge sources and producing the full search network optimized up to the HMM states[8]. We combined cross-word triphones and trigrams into a single WFST, and applied a one-pass search algorithm to the WFST. The confidence measure for each word was calculated by post-processing.

Screening filter

The transcribed question by an ASR system sometimes includes not only redundant information caused by the spontaneity of human speech but also irrelevant information caused by recognition errors. In order to extract meaningful information, recognition errors, fillers, word fragments and so on are removed from the transcribed question by a screening filter. The summarization method [9] is applied to the screening process. In this approach, a set of words maximizing a summarization score indicating the appropriateness of summarization is extracted from automatically a transcribed question and these words are then concatenated together. The extraction process is performed using a Dynamic Programming (DP) technique.

Since recognition errors in recognition results degrade the QA performance directly, the screening filter should remove such recognition errors. In this study, the screening process is performed with 2 steps. The first step is to remove acoustically and linguistically unreliable words based on the threshold of the confidence measure. The second step is to construct a meaningful sentence from the results after removing the unreliable words using a speech summarization technique [9]. Hence the screened results excludes the large recognition errors and becomes understandable sentence. Finally, the screened result is input into the ODQA engine.

ODQA engine

The ODQA engine consists of four components: question analysis, text retrieval, answer hypotheses extraction, and answer selection. Nouns/noun-phrases are classified into category classes such as ORGANIZATION or PERSON. A given question sentence is analyzed to determine its expected answer type and keywords by the question analysis module. And then paragraphs/documents that match the keywords are extracted by the text retrieval module. The nouns/noun-phrases in the retrieved relevant documents that belong to the expected category class are extracted and used to output answers.

DDQ module

When the ODQA engine cannot extract an appropriate answer for a user's question, the question is considered "ambiguous." There are two cases in which a question becomes ambiguous. One is the case where the user does not present enough information in the question. The other is the case where some information is lost through the ASR. In such cases, the DDQ module derives disambiguous queries (DQs) that require additional information to be given so that lacking information can be supplemented and the correct answer can be distinguished.

The DQs are generated by using templates of interrogative sentences, each of which contains an interrogative and a phrase taken from a user's question after speech recognition and screening. The DDQ module selects the best DQ based on its linguistic appropriateness and ambiguity of the phrase. Hence, the module can generate a sentence which is linguistically appropriate and ask the user to disambiguate the most ambiguous phrase in his/her question.

Suppose the DDQ module receives the question sentence:

Which country in South Africa won the world cup?

If the phrase "the world cup" is considered to be ambiguous, it is effective to ask the user to supplement information about "the world cup" such as the event (i.e. soccer, volley ball), the venue, the season, etc. For example, the following DQs are hypothesized by inserting the ambiguous phrase into the templates.

What kind of world cup?

What year was the world cup held?

Where is South Africa?

The linguistic appropriateness of DQs can be measured by using a language model such as a trigram. The ambiguity of each phrase is measured by using a structural ambiguity and a generality score for the phrase.

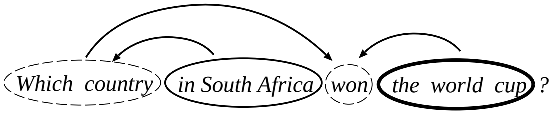

Fig. 2. An example of dependency structure.

The structural ambiguity is based on the dependency structure of a sentence. A phrase that is not modified by the other phrases is considered to have a high ambiguity. Figure 2 shows an example of a dependency structure, in which the sentence is separated into phrases. Each arrow represents a dependency between two phrases. In this case, no phrases modify "the world cup." We assume that ambiguity for such a phrase is higher than for the others. The structural ambiguity of n-th phrase is defined as

$$A_D(P_n) = \log \left\{ 1 - \sum_{i=1:i \neq n}^{N} D(P_i, P_n) \right\},$$

where, the question sentence is separated into N phrases, and $D(P_i, P_n)$ is a probability that the phrase P_n is modified by the phrase P_i, which can be calculated using a Stochastic Dependency Context Free Grammar (SDCFG) [11].

In addition, the generality score of a phrase is also incorporated to measure the ambiguity of noun/noun-phrases. Nouns/noun-phrases that occur frequently in a corpus rarely help answer extraction. We assume that such a phrase is ambiguous and should be modified by additional information. The generality score is defined as

$$A_G(P_n) = \sum_{w \in P_n : w = \text{cont}} \log P(w),$$

where $P(w)$ is a unigram probability of w based on a corpus to be retrieved. "w = cont" means that w is a content word such as noun, verb, adjective, and so on.

Let S_{mn} be a DQ generated by inserting the n-th phrase into the m-th template. The DDQ module selects the DQ which maximizes the DQ score:

$$H(S_{mn}) = \lambda_L L(S_{mn}) + \lambda_D A_D(P_n) + \lambda_G A_G(P_n),$$

where $L(\cdot)$ is a linguistic score such as the logarithm of the trigram probability. λ_L, λ_D, and λ_G are weighting factors for balancing of the scores.

Our system is actually built for Japanese speech. Japanese sentences can be divided into phrase-like units (*bunsetsu*). The phrase-like unit *bunsetsu* is denoted by 'phrase'. Since a new phrase starts always from a content word, a sentence is split into a phrase sequence based on the first content word. Each phrase is made up of a content word followed by zero or more function words, and each word modifies succeeding words within the phrase. In addition, since Japanese sentences have only "right-headed" dependency, the dependency probability $D(P_k, P_l)$ is 0 if $k \geq l$.

3. EVALUATION EXPERIMENTS

Questions consisting of 69 sentences read aloud by seven male speakers were transcribed by our ASR system [8]. These questions were prepared to test the performance of our ODQA engine [10]. Each question consists of about 19 morphemes on average. The sentences are grammatically correct, formally structured and have enough information for the ODQA engine to obtain the exact answers. Therefore, transcription results with 100% word accuracy can accomplish answer extraction accurately. On the contrary, transcription results including recognition errors fail to extract correct answers. Mean word recognition accuracies of 69 questions were 76%. The question transcriptions were processed by the screening filter and input into the ODQA engine. The DDQ module generated DQs based on the screened results.

3.1. ASR system

The speech signal was sampled at 16kHz with 16 bit quantization. Feature vectors had 25 elements consisting of 12 MFCC, their delta, and delta log energies. Tied-state triphone HMMs with 3000 states and 16 Gaussians per state were made by using 338 spontaneous presentations uttered by male speakers (approximately 59 hours). Decoding was done by a one-pass Viterbi search using a WFST, integrating cross-word triphone HMMs and trigrams [8].

3.2. Screening filter

The screening process was performed by removing recognition errors using a confidence measure as threshold and then summarizing it within 80% to 100% compaction ratio. In this summarization technique [9], the word significance score and the linguistic score for summarization were calculated using text from Mainichi newspaper published from 1994 to 2001, comprising of 13.6M sentences with 232M words. SDCFG for word concatenation score was the same as that used in [9]. The posterior probability of each transcribed word in a word graph obtained by the ASR system was used as the confidence score.

3.3. DDQ module

The word generality score A_G was computed using the same set of text from Mainichi newspaper that was used for the screening filter. Eighty-two kinds of interrogative sentences were created as disambiguous queries for each noun/noun-phrase in each question and evaluated in the DDQ module. The linguistic score L indicating appropriateness of interrogative sentences was calculated using 1000 questions and newspaper text for three years. The structural ambiguity score $A_D$was calculated based on the SDCFG which was used for the screening filter.

3.4. Evaluation method

The DQs generated by the DDQ module were evaluated in comparison with manual disambiguation queries. Since the questions read by the seven speakers had no redundancy for obtaining exact answers, every recognition error resulted in loss of information indispensable for obtaining the correct answers. The manual DQs were made by five human subjects based on the comparison of the original written questions and the transcription results given by the ASR system. The automatic DQs were categorized into three classes: APPROPRIATE when they had the same meaning as at least one of the five manual DQs, InAPPROPRIATE when they had no match, and HELPFUL when the meanings were partially matched.

3.5. Evaluation results

Table 2 shows the evaluation results in terms of the three categories. These results indicate that roughly 50% of the DQs generated by the DDQ module based on the recognition results were APPROPRIATE, which means that the DDQ module effectively generated queries to disambiguate the users' questions.

Table 2. Evaluation results of disambiguous queries generated by the DDQ module.

Speaker	Word accuracy	Sent. w/o errors	APP	Helpful	InAPP
A	70%	4	32	5	28
B	76%	8	36	3	22
C	79%	10	34	1	24
D	73%	4	35	2	28
E	78%	7	31	2	29
F	80%	8	34	2	25
G	74%	3	35	3	28
Mean	76%	9%	49%	4%	38%

A number without a % indicates number of sentences.

4. CONCLUSION

This paper has proposed a new strategy for spoken interactive ODQA (open-domain question answering) systems. In this strategy, when a user's question is ambiguous, additional information indispensable for extracting an exact answer is automatically queried by the DDQ (deriving disambiguous queries) module. The DDQ module generates a DQ (disambiguous query) using an ambiguous phrase in the user's question extracted based on the structural ambiguity of the question and the generality of the phrase. Experimental results show that generated DQs were effective in requiring lacking information caused by speech recognition errors.

Future research includes evaluation of the proposed strategy by the performance of the total QA system from the viewpoint of how much the total performance is improved by using the DQs.

5. REFERENCES

[1] http://trec.nist.gov

[2] F. Pereira et. al., "Definite Clause Grammars for Language Analysis –a Survey of the Formalism and a Comparison with Augmented Transition Networks," *Artificial Intelligence*, 13:231-278, 1980.

[3] S. Harabagiu et. al., "Experiments with Open-Domain Textual Question Answering," *COLING-2000*, pages 292-298, Saarbruken Germany, August 2000.

[4] E. H. Shortliffe, "Computer-Based Medical Consultations: MYCIN," *Elsevier/North Holland*, New York NY, 1976.

[5] T. Lowerre et. al., "The Harpy speech understanding system," W. A. Lea (Ed.), *Trends in Speech recognition*, pp. 340, Prentice Hall.

[6] V. Zue, et al., "JUPITER: A Telephone-Based Conversational Interface for Weather Information," *IEEE Transactions on Speech and Audio Processing*, Vol. 8 , No. 1, 2000.

[7] S. Harabagiu et. al., "Expanding the scope of the ATIS Task: the ATIS-3 Corpus," *COLING2002*, vol.I, pp.321–327, Taipei, 1994.

[8] D. Willett et. al., "Time and memory efficient Viterbi decoding for LVCSR using a precompiled search network," *Proc. of Eurospeech 2001*, vol.2, pp.847–850, 2001.

[9] C. Hori et.al., "A New Approach to Automatic Speech Summarization," To appear in the *IEEE Transactions on Multimedia*, 2002.

[10] Y. Sasaki et. al., "NTT's QA Systems for NTCIR QAC-1," *Proc. of NTCIR Workshop Meeting*, pp.63–70, 2000.

[11] C. Hori et.al., "A Statistical Approach for Automatic Speech Summarization," To appear in the *EURASIP Journal on Applied Signal Processing*, 2003.