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 extract an exact answer from large text corpora based on text input are intensively being investigated. However, the information in the first question input by a user is not usually enough to yield the desired answer. Interactions for collecting additional information to accomplish QA is needed. This paper proposes an interactive approach for spoken interactive ODQA systems. When the reliabilities for answer hypotheses obtained by an ODQA system are low, 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 to supplementing a lack of 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. We confirmed the appropriateness of the derived DQs by comparing them with manually prepared ones.

1. INTRODUCTION

In the field of spoken language processing (SLP), human and machine dialogue systems using a speech interface have been intensively researched. Some of these systems are marketed in phone systems, i.e., for air ticket reservations or information retrieval for stock prices. Such conversational dialogues to 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). To achieve more natural communication between human beings and machines, spoken dialogue systems for open domains are necessary. Specifically, 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. To clarify the problems presented in building such a system, the QA systems that have been constructed so far have been classified into a number of groups depending on their target domains, interfaces, and interactions to draw out additional information from users to accomplish set tasks shown in Table 1. In this table, text and speech denote text input and speech input, respectively. The term "*addition*" represents additional information queried by the QA systems. This additional information is other than that derived from the user's initial questions.

ODQA that extracts answers from large text corpora, such as newspaper texts, has been intensively investigated in the field of natural language processing (NLP). The Text REtrieval Conference (TREC), co-sponsored by the NIST and DARPA, has had an ODQA track since 1999 (TREC-8) [1]. Although the ODQA task is an Information Retrieval (IR) issue, the ODQA systems return an actual answer rather than a ranked list of documents in response to a question written in a natural language. However, SDQA has been researched in the area of artificial intelligence (AI). The differences between SDQA and ODQA systems are in their data structures and the design of dialogue scenarios. Since information in a specific domain can be arranged in a table, the SDQA systems such as CHAT-80 [2] can accomplish QA by table lookup techniques. However, since information in an open domain is scattered in large unstructured text corpora, the table-look-up technique cannot be applied.

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.

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

To obtain more exact answers to questions, some QA systems have interactions with users that can capture additional information to accomplish their 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-

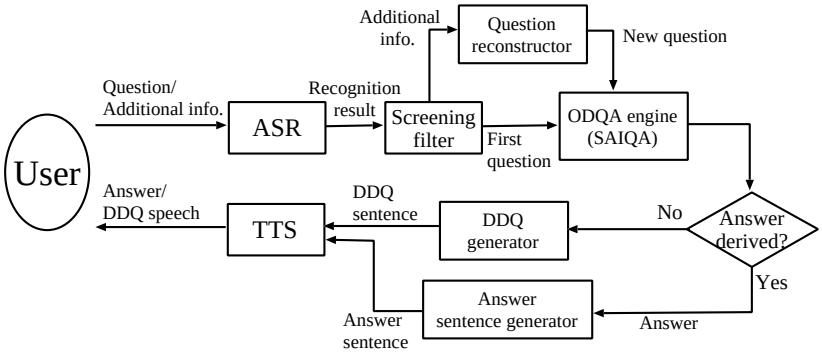

Fig. 1. Components and data flow in SPIQA.

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

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

To construct more precise and user-friendly ODQA systems, this paper proposes an interactive approach to spoken ODQA systems. Three main issues that need to be addressed to construct spoken interactive ODQA systems are:

1. The ODQA problems:
   Answers are not in a table and are scattered throughout unstructured text.

2. The interactive ODQA problems:
   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 the user inputs the question.

3. The spoken QA problems:
   Recognition errors degrade the performance of QA systems. Some indispensable information to extract answers is deleted or substituted by other words.

This paper proposes an interactive approach based on disambiguation of users' questions in interactive ODQA systems. In addition, we introduce our spoken interactive ODQA system, i.e., **SPIQA**.

2. SPOKEN INTERACTIVE QA SYSTEM: SPIQA

Figure 1 shows the components of our system, and the data that flows through it. This system is comprised of an ASR system [8], a screening filter that uses a summarization method [9], an ODQA engine (**SAIQA**) [10] for a Japanese newspaper text corpus, and a Deriving Disambiguous Queries (**DDQ**) module.

ASR system

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

Screening filter

The question transcribed by an ASR system sometimes incorporates not only redundant information caused by the spontaneity of human speech but also irrelevant information due to recognition errors. Recognition errors, fillers, word fragments, and other distractors are removed from the transcribed question by a screening filter that extracts meaningful information. the summarization method [9] is applied to the screening process.

Since recognition errors in the recognition results directly degrade QA performance, the screening filter should remove them. In this study, the screening process was done in two steps. The first step was to remove acoustically and linguistically unreliable words based on the threshold for the confidence measure. The second step was to construct a meaningful sentence from the results after removing unreliable words using the speech summarization technique through word extraction [9]. Hence, the screened results excluded large recognition errors and made the sentence understandable. Finally, the screened results were input into the ODQA engine.

ODQA engine

The ODQA engine has four components: question analysis, text retrieval, answer hypothesis 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 the type of expected answer and keywords

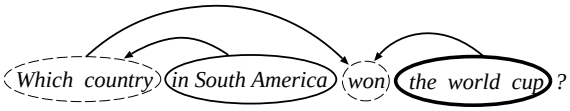

Fig. 2. Example of dependency structure.

using the question analysis module. Paragraphs/documents that match the keywords are then 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 to a user's question, the question is considered "ambiguous." There are two situations where the question is considered ambiguous. The first is when the user does not supply sufficient information in his/her question. The other is when some necessary information to extract the answer is lost through ASR. Since all information in a users' question is not always useful to extract answers, only indispensable information to do this should be compensated by additional information that is inputs by users. The DDQ module derives disambiguous queries (DQs) that require such indispensable information.

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

Suppose the DDQ module is posed with this question:

Which country in South America won the World Cup?

If the phrase "the World Cup" is considered ambiguous, it is necessary to ask the user to supplement information corresponding to "the World Cup" such as the name of the sport (i.e. soccer, volleyball), the venue, the season, and other characteristics. For example, the following DQs can be hypothesized by inserting an ambiguous phrase into the templates.

What kind of World Cup?
What year was the World Cup held?
Where is South America?

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 the structural ambiguity and generality score for the phrase.

The structural ambiguity is based on the dependency structure of the sentence. A phrase that is not modified by other phrases is considered to be highly ambiguous. Figure 2 has an example of a dependency structure, where the question is separated into phrases. Each arrow represents the dependency between two phrases. Here, no phrases modify "the World Cup." We assume that ambiguity for such a phrase would be higher than for others. The structural ambiguity of the 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 complete question is separated into N phrases, and $D(P_i, P_n)$ is the probability that phrase P_n will be modified by phrase P_i, which can be calculated using Stochastic Dependency Context Free Grammar (SDCFG) [11].

In addition, the generality score of a phrase is also incorporated into measuring the ambiguity of noun/noun-phrases. Nouns/noun-phrases that frequently occur in a corpus rarely help to extract answers. 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 the unigram probability of w based on the corpus to be retrieved. "$w =$ cont" means that w is a content word such as a noun, verb or adjective.

Let S_{mn} be a DQ generated by inserting the n-th phrase into the m-th template. The DDQ module selects the DQ that 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 for trigram probability. λ_L, λ_D, and λ_G are weighting factors to balance 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 always starts 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 consisted of about 19 morphemes on average. The sentences were grammatically correct, formally structured, and had enough information for the ODQA engine to extract the correct answers. Therefore, transcription results with 100% word accuracy could extract answers accurately. In contrast, transcription results with recognition errors failed to extract correct answers. The mean word recognition accuracy of 69 questions was 76%. The question transcriptions were processed with a screening filter and input into the ODQA engine.

3.1. ASR system

The speech signal was sampled at 16 kHz 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 prepared by using 338 spontaneous presentations uttered by male speakers (approximately 59 hours). Decoding was done with a one-pass Viterbi search using WFST, integrating cross-word triphone HMMs and trigrams [8].

3.2. Screening filter

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

3.3. DDQ module

The word generality score A_G was computed using the same Mainichi newspaper text that was used for screening. 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 the appropriateness of interrogative sentences was calculated using 1000 questions and newspaper text extracted 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. Although the questions read by the seven speakers had sufficient information to extract exact answers, some recognition errors resulted in a loss of information that was indispensable for obtaining the correct answers. The manual DQs were made by five subjects based on a 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 there was no match, and HELPFUL when the meanings were partially matched. QA performance using recognition results was evaluated by the MRR (Mean Reciprocal Rank) [12].When the correct answer for each question was included within the top five answers given by the ODQA system, the answer was judged to be correct, and its reciprocal rank was accumulated.When QA systems outputted perfect answers, the MRRs was 1.0. The higher MRRs indicated that QA performance was higher.

3.5. Evaluation results

Table 2 shows the evaluation results in terms of the appropriateness of the DQs and the QA-system MRR. The mean MRRs for manual transcription was 0.43. However, the mean MRRs for the recognition results was 0.27. The 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 for disambiguating the user's questions. In addition, speakers with lower recognition accuracies produced lower MRRs, that is, a lack of information due to recognition errors caused lower QA performance. Experimental results revealed the potential of the generated DQs in compensating the degradation of the QA performance due to recognition errors.

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

Speaker	Word acc.	MRR	Sent. w/o errors	APP	Help -ful	InAPP
A	70%	0.20	4	32	5	28
B	76%	0.28	8	36	3	22
C	79%	0.27	10	34	1	24
D	73%	0.30	4	35	2	28
E	78%	0.28	7	31	2	29
F	80%	0.31	8	34	2	25
G	74%	0.20	3	35	3	28
Mean	76%	0.26	9%	49%	4%	38%

An integer without a % indicates number of sentences.

4. CONCLUSION

This paper 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 to extract the 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 that was extracted based on the structural ambiguity of the question and the generality of the phrase. Experimental results revealed the potential of the generated DQs in requiring indispensable information that was lacking to extract answers. Future research will include an evaluation of the strategy using the performance of the total QA system based on how much the DQs improve the total performance.

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*, pp. 292-298, Saarbrüken 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., "Open-Domain Voice-Activated Question Answering," *COLING2002*, vol. I, pp. 321–327, Taipei, 2002.

[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*, 2003.

[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.

[12] http://trec.nist.gov/data/qa.html