Towards Conversational QA: Automatic Identification of Problematic Situations and User Intent [*]

Joyce Y. Chai **Chen Zhang** **Tyler Baldwin**
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
{jchai, zhangch6, baldwi96}@cse.msu.edu

Abstract

To enable conversational QA, it is important to examine key issues addressed in conversational systems in the context of question answering. In conversational systems, understanding user intent is critical to the success of interaction. Recent studies have also shown that the capability to automatically identify problematic situations during interaction can significantly improve the system performance. Therefore, this paper investigates the new implications of user intent and problematic situations in the context of question answering. Our studies indicate that, in basic interactive QA, there are different types of user intent that are tied to different kinds of system performance (e.g., problematic/error free situations). Once users are motivated to find specific information related to their information goals, the interaction context can provide useful cues for the system to automatically identify problematic situations and user intent.

1 Introduction

Interactive question answering (QA) has been identified as one of the important directions in QA research (Burger et al., 2001). One ultimate goal is to support intelligent conversation between a user and a QA system to better facilitate user information needs. However, except for a few systems that use dialog to address complex questions (Small et al., 2003; Harabagiu et al., 2005), the general dialog capabilities have been lacking in most question answering systems. To move towards conversational QA, it is important to examine key issues relevant to conversational systems in the context of interactive question answering.

This paper focuses on two issues related to conversational QA. The first issue is concerned with user intent. In conversational systems, understanding user intent is the key to the success of the interaction. In the context of interactive QA, one question is what type of user intent should be captured. Unlike most dialog systems where user intent can be characterized by dialog acts such as *question*, *reply*, and *statement*, in interactive QA, user inputs are already in the form of *question*. Then the problems become whether there are different types of intent behind these questions that should be handled differently by a QA system and how to automatically identify them.

The second issue is concerned with problematic situations during interaction. In spoken dialog systems, many problematic situations could arise from insufficient speech recognition and language understanding performance. Recent work has shown that the capability to automatically identify problematic situations (e.g., speech recognition errors) can help control and adapt dialog strategies to improve performance (Litman and Pan, 2000). Similarly, QA systems also face challenges of technology limitation from language understanding and information retrieval. Thus one question is, in the context of interactive QA, how to characterize problematic situations and automatically identify them when they occur.

In interactive QA, these two issues are intertwined. Questions formed by a user not only depend on his/her information goals, but are also influenced by the answers from the system. Problematic situations will impact user intent in the

[*]This work was partially supported by IIS-0347548 from the National Science Foundation.

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 57–64,
Sydney, July 2006. ©2006 Association for Computational Linguistics

follow-up questions, which will further influence system performance. Both the awareness of problematic situations and understanding of user intent will allow QA systems to adapt better strategies during interaction and move towards intelligent conversational QA.

To address these two questions, we conducted a user study where users interacted with a *controlled* QA system to find information of interest. These controlled studies allowed us to focus on the interaction aspect rather than information retrieval or answer extraction aspects. Our studies indicate that in *basic interactive QA* where users always ask questions and the system always provides some kind of answers, there are different types of user intent that are tied to different kinds of system performance (e.g., problematic/error free situations). Once users are motivated to find specific information related to their information goals, the interaction context can provide useful cues for the system to automatically identify problematic situations and user intent.

2 Related Work

Open domain question answering (QA) systems are designed to automatically locate answers from large collections of documents to users' natural language questions. In the past few years, automated question answering techniques have advanced tremendously, partly motivated by a series of evaluations conducted at the Text Retrieval Conference (TREC) (Voorhees, 2001; Voorhees, 2004). To better facilitate user information needs, recent trends in QA research have shifted towards complex, context-based, and interactive question answering (Voorhees, 2001; Small et al., 2003; Harabagiu et al., 2005). For example, NIST initiated a special task on context question answering in TREC 10 (Voorhees, 2001), which later became a regular task in TREC 2004 (Voorhees, 2004) and 2005. The motivation is that users tend to ask a sequence of related questions rather than isolated single questions to satisfy their information needs. Therefore, the context QA task was designed to investigate the system capability to track context through a series of questions. Based on context QA, some work has been done to identify clarification relations between questions (Boni and Manandhar, 2003). However context QA is different from interactive QA in that context questions are specified ahead of time rather than incrementally as in an interactive setting.

Interactive QA has been applied to process complex questions. For analytical and non-factual questions, it is hard to anticipate answers. Clarification dialogues can be applied to negotiate with users about the intent of their questions (Small et al., 2003). Recently, an architecture for interactive question answering has been proposed based on a notion of predictive questioning (Harabagiu et al., 2005). The idea is that, given a complex question, the system can automatically identify a set of potential follow-up questions from a large collection of question-answer pairs. The empirical results have shown the system with predictive questioning is more efficient and effective for users to accomplish information seeking tasks in a particular domain (Harabagiu et al., 2005).

The work reported in this paper addresses a different aspect of interactive question answering. Both issues raised earlier (Section 1) are inspired by earlier work on intelligent conversational systems. Automated identification of user intent has played an important role in conversational systems. Tremendous amounts of work has focused on this aspect (Stolcke et al., 2000). To improve dialog performance, much effort has also been put on techniques to automatically detect errors during interaction. It has shown that during human machine dialog, there are sufficient cues for machines to automatically identify error conditions (Levow, 1998; Litman et al., 1999; Hirschberg et al., 2001; Walker et al., 2002). The awareness of erroneous situations can help systems make intelligent decisions about how to best guide human partners through the conversation and accomplish the tasks. Motivated by these earlier studies, the goal of this paper is to investigate whether these two issues can be applied in question answering to facilitate intelligent conversational QA.

3 User Studies

We conducted a user study to collect data concerning user behavior in a basic interactive QA setting. We are particularly interested in how users respond to different system performance and its implication in identifying problematic situations and user intent. As a starting point, we characterize system performance as either *problematic*, which indicates the answer has some problem, or *error-free*, which indicates the answer is correct. In this section, we first describe the methodology and the system used in this effort and then discuss the observed user behavior and its relation to problematic situations and user intent.

3.1 Methodology and System

The system used in our experiments has a user interface that takes a natural language question and presents an answer passage. Currently, our interface only presents to the user the top one retrieved result. This simplification on one hand helps us focus on the investigation of user responses to different system performances and on the other hand represents a possible situation where a list of potential answers may not be practical (e.g., through PDA or telephone line).

We implemented a Wizard-of-Oz (WOZ) mechanism in the interaction loop to control and simulate problematic situations. Users were not aware of the existence of this human wizard and were led to believe they were interacting with a real QA system. This *controlled* setting allowed us to focus on the interaction aspect rather than information retrieval or answer extraction aspect of question answering. More specifically, during interaction after each question was issued, a random number generator was used to decide if a problematic situation should be introduced. If the number indicated no, the wizard would retrieve a passage from a database with correct question/answer pairs. Note that in our experiments we used specific task scenarios (described later), so it was possible to anticipate user information needs and create this database. If the number indicated that a problematic situation should be introduced, then the Lemur retrieval engine [1] was used on the AQUAINT collection to retrieve the answer. Our assumption is that AQUAINT data are not likely to provide an exact answer given our specific scenarios, but they can provide a passage that is most related to the question. The use of the random number generator was to control the ratio between the occurrence of problematic situations and error-free situations. In our initial investigation, since we are interested in observing user behavior in problematic situations, we set the ratio as 50/50. In our future work, we will vary this ratio (e.g., 70/30) to reflect the performance of state-of-the-art factoid QA and investigate the implication of this ratio in automated performance assessment.

3.2 Experiments

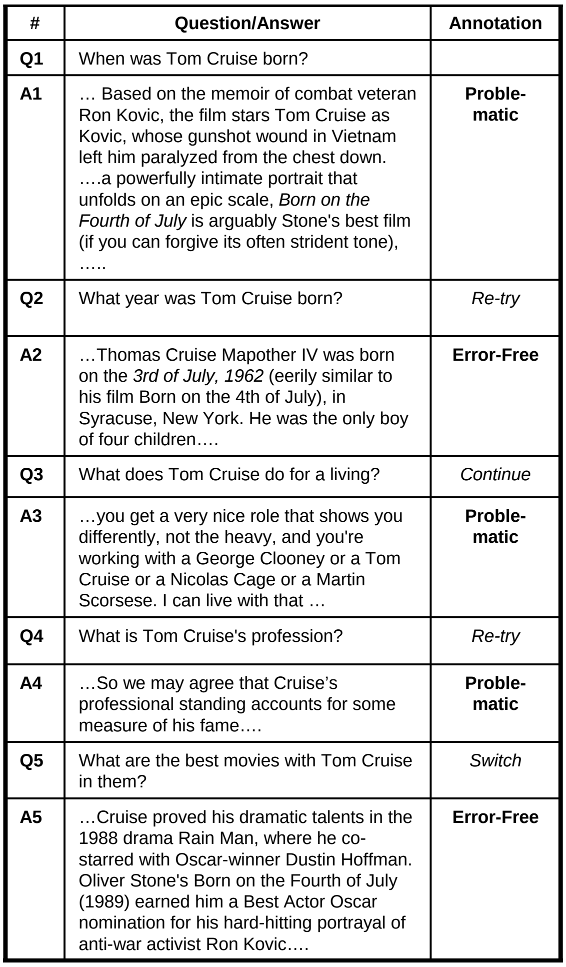
Eleven users participated in our study. Each user was asked to interact with our system to complete information seeking tasks related to four specific scenarios: *the 2004 presidential debates*, *Tom Cruise*, *Hawaii*, and *Pompeii*. The experimental scenarios were further divided into two types: structured and unstructured. In the structured task scenarios (for topics *Tom Cruise* and *Pompeii*), users had to fill in blanks on a diagram pertaining to the given topic. Using the diagram was to avoid the influence of these scenarios on the language formation of the relevant questions. Because users must find certain information, they were constrained in the range of questions in which they could ask, but not the way they ask those questions. The task was completed when all of the blanks on the diagram were filled. The structured scenarios were designed to mimic the real information seeking practice in which users have real motivation to find specific information related to their information goals. In the unstructured scenarios (for topics *the 2004 presidential debates* and *Hawaii*), users were given a general topic to investigate, but were not required to find specific information. This gave the user the ability to ask a much wider range of questions than the structured scenarios. Users were generally in an exploration mode when performing these unstructured tasks. They were not motivated to find specific information and were content with any information provided by the system. In our view, the unstructured scenarios are less representative of the true information seeking situations.

3.3 Observations and Analysis

From our studies, a total of 44 interaction sessions with 456 questions were collected. Figure 1 shows an example of a fragment of interaction related to *Tom Cruise*. In this example, both problematic situations applied to answers (e.g., *Problematic* and *Error-Free*) and user intent (described later) applied to questions are annotated.

There are several observations from this data. First, questions formed during interactive QA tend to be self-contained and free of definite noun phrases, pronouns, or ellipsis. Only one question in the entire data set has a pronoun (i.e., *What are the best movies with Tom Cruise in them?*). Even in this case, the pronoun *them* did not refer to any entities that occurred previously in the

[1] http://www-2.cs.cmu.edu/ lemur/



#	Question/Answer	Annotation
Q1	When was Tom Cruise born?	
A1	… Based on the memoir of combat veteran Ron Kovic, the film stars Tom Cruise as Kovic, whose gunshot wound in Vietnam left him paralyzed from the chest down. ….a powerfully intimate portrait that unfolds on an epic scale, *Born on the Fourth of July* is arguably Stone's best film (if you can forgive its often strident tone), …..	**Problematic**
Q2	What year was Tom Cruise born?	*Re-try*
A2	…Thomas Cruise Mapother IV was born on the *3rd of July, 1962* (eerily similar to his film Born on the 4th of July), in Syracuse, New York. He was the only boy of four children….	**Error-Free**
Q3	What does Tom Cruise do for a living?	*Continue*
A3	…you get a very nice role that shows you differently, not the heavy, and you're working with a George Clooney or a Tom Cruise or a Nicolas Cage or a Martin Scorsese. I can live with that …	**Problematic**
Q4	What is Tom Cruise's profession?	*Re-try*
A4	…So we may agree that Cruise's professional standing accounts for some measure of his fame….	**Problematic**
Q5	What are the best movies with Tom Cruise in them?	*Switch*
A5	…Cruise proved his dramatic talents in the 1988 drama Rain Man, where he co-starred with Oscar-winner Dustin Hoffman. Oliver Stone's Born on the Fourth of July (1989) earned him a Best Actor Oscar nomination for his hard-hitting portrayal of anti-war activist Ron Kovic….	**Error-Free**

Figure 1: An example fragment of interaction

QA process. This phenomenon could be caused by how the answers are presented. Unlike specific answer entities, the answer passages provided by our system do not support the natural use of referring expressions in the follow-up questions. Another possible explanation could be that in an interactive environment, users seem to be more aware of the potential limitation of a computer system and thus tend to specify self-contained questions in a hope to reduce the system's inference load.

The second observation is about user behavior in response to different system performances (i.e., problematic or error-free situations). We were hoping to see different strategies users might apply to deal with the problematic situations. However, based on the data, we found that when a problem occurred, users either rephrased their questions (i.e., the same question expressed in a different way) or gave up the question and went on specifying a new question. (Here we use *Rephrase* and *New* to denote these two kinds of behaviors.) We have not observed any sub-dialogs initiated by the user to clarify a previous question or answer. One possible explanation is that the current investigation was conducted in a basic interactive mode where the system was only capable of providing some sort of answers. This may limit users' expectation in the kind of questions that can be handled by the system. Our assumption is that, once the QA system becomes more intelligent and able to carry on conversation, different types of questions (i.e., other than rephrase or new) will be observed. This hypothesis certainly needs to be validated in a conversational setting.

	Problematic	**Error-free**	**Total**
New	*Switch*	*Continue*	
unstruct.	29	90	119
struct.	29	133	162
entire	58	223	281
Rephrase	*Re-try*	*Negotiate*	
unstruct.	19	4	23
struct.	102	6	108
entire	121	10	131
Total-unst	48	94	142
Total-st	131	139	270
Total-ent	179	233	412

Table 1: Categorization of user intent with the corresponding number of occurrences from the unstructured scenarios, the structured scenarios, and the entire dataset.

The third observation is that the rephrased questions seem to strongly correlate with problematic situations, although not always. New questions cannot distinguish a problematic situation from an error-free situation. Table 1 shows the statistics from our data about different combinations of new/rephrase questions and performance situations[2]. What is interesting is that these different combinations can reflect different types of user intent behind the questions. More specifically, given a question, four types of user intent can be captured with respect to the context (e.g., the previous question and answer)

Continue indicates that the user is satisfied with the previous answer and now moves on to this new question.

Switch indicates that the user has given up on the previous question and now moves on to this

[2] The last question from each interaction session is not included in these statistics because there is no follow-up question after that.

new question.

Re-try indicates that the user is not satisfied with the previous answer and now tries to get a better answer.

Negotiate indicates that the user is not satisfied with the previous answer (although it appears to be correct from the system's point of view) and now tries to get a better answer for his/her own needs.

Table 1 summarizes these different types of intent together with the number of corresponding occurrences from both structured and unstructured scenarios. Since in the unstructured scenarios it was hard to anticipate user's questions and therefore take a correct action to respond to a problematic/error-free situation, the distribution of these two situations is much more skewed than the distribution for the structured scenarios. Also as mentioned earlier, in unstructured scenarios, users lacked the motivation to pursue specific information, so the ratio between *switch* and *re-try* is much larger than that observed in the structured scenarios. Nevertheless, we did observe different user behavior in response to different situations. As discussed later in Section 5, identifying these fine-grained intents will allow QA systems to be more proactive in helping users find satisfying answers.

4 Automatic Identification of Problematic Situations and User Intent

Given the discussion above, the next question is how to automatically identify problematic situations and user intent. We formulate this as a classification problem. Given a question Q_i, its answer A_i, and the follow-up question Q_{i+1}:

(1) Automatic identification of problematic situations is to decide whether A_i is problematic (i.e., correct or incorrect) based on the follow-up question Q_{i+1} and the interaction context. This is a binary classification problem.

(2) Automatic identification of user intent is to identify the intent of Q_{i+1} given the interaction context. Because we only have very limited instances of *Negotiate* (see Table 1), we currently merge *Negotiate* with *Re-try* since both of them represent a situation where a better answer is requested. Thus, this problem becomes a trinary classification problem.

To build these classifiers, we identified a set of features, which are illustrated next.

4.1 Features

Given a question Q_i, its answer A_i, and the follow-up question Q_{i+1}, the following set of features are used:

Target matching(TM): a binary feature indicating whether the target type of Q_{i+1} is the same as the target type of Q_i. Our data shows that the repetition of the target type may indicate a rephrase, which could signal a problematic situation has just happened.

Named entity matching (NEM): a binary feature indicating whether all the named entities in Q_{i+1} also appear in Q_i. If no new named entity is introduced in Q_{i+1}, it is likely Q_{i+1} is a rephrase of Q_i.

Similarity between questions (SQ): a numeric feature measuring the similarity between Q_{i+1} and Q_i. Our assumption is that the higher the similarity is, the more likely the current question is a rephrase to the previous one.

Similarity between content words of questions (SQC): this feature is similar to the previous feature (i.e., SQ) except that the similarity measurement is based on the content words excluding named entities. This is to prevent the similarity measurement from being dominated by the named entities.

Similarity between Q_i and A_i (SA): this feature measures how close the retrieved passage matches the question. Our assumption is that although a retrieved passage is the most relevant passage compared to others, it still may not contain the answer (e.g., when an answer does not even exist in the data collection).

Similarity between Q_i and A_i based on the content words (SAC): this feature is essentially the same as the previous feature (SA) except that the similarity is calculated after named entities are removed from the questions and answers.

Note that since our data is currently collected from simulation studies, we do not have the confidence score from the retrieval engine associated with every answer. In practice, the confidence score can be used as an additional feature.

Since our focus is not on the similarity measurement but rather the use of the measurement in the classification models, our current similarity measurement is based on a simple approach that measures commonality and difference between two objects as proposed by Lin (1998). More specifically, the following equation is applied to measure the similarity between two chunks of text T_1 and T_2:

$$sim_1(T_1, T_2) = \frac{-\log P(T_1 \cap T_2)}{-\log P(T_1 \cup T_2)}$$

Assume the occurrence of each word is independent, then:

$$sim_1(T_1, T_2) = \frac{-\sum_{w \in T_1 \cap T_2} \log P(w)}{-\sum_{w \in T_1 \cup T_2} \log P(w)}$$

where $P(w)$ was calculated based on the data used in the previous TREC evaluations.

4.2 Identification of Problematic Situations

To identify problematic situations, we experimented with three different classifiers: Maximum Entropy Model (MEM) from MALLET[3], SVM from SVM-Light[4], and Decision Trees from WEKA[5]. A leave-one-out validation was applied where one interaction session was used for testing and the remaining interaction sessions were used for training.

Table 2 shows the performance of the three models based on different combinations of features in terms of classification accuracy. The baseline result is the performance achieved by simply assigning the most frequently occurred class. For the unstructured scenarios, the performance of the classifiers is rather poor, which indicates that it is quite difficult to make any generalization based on the current feature sets when users are less motivated in finding specific information. For the structured scenarios, the best performance for each model is highlighted in bold in Table 2. The Decision Tree model achieves the best performance of 77.8% in identifying problematic situations, which is more than 25% better than the baseline performance.

4.3 Identification of User Intent

To identify user intent, we formulate the problem as follows: given an observation feature vector **f** where each element of the vector corresponds to a feature described earlier, the goal is to identify an intent c^* from a set of intents I ={*Continue, Switch, Re-try/Negotiate*} that satisfies the following equation:

$$c^* = \arg\max_{c \in I} P(c|\mathbf{f})$$

Our assumption is that user intent for a question can be potentially influenced by the intent from a preceding question. For example, *Switch* is likely to follow *Re-try*. Therefore, we have implemented a Maximum Entropy Markov Model (MEMM) (McCallum et al., 2000) to take the sequence of interactions into account.

Given a sequence of questions Q_1, Q_2, up to Q_t, there is an observation feature vector $\mathbf{f_i}$ associated with each Q_i. In MEMM, the prediction of user intent c_t for Q_t not only depends on the observation $\mathbf{f_t}$, but also the intent c_{t-1} from the preceding question Q_{t-1}. In fact, this approach finds the best sequence of user intent C^* for Q_1 up to Q_t based on a sequence of observations $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_t$ as follows:

$$C^* = \arg\max_{C \in I^t} P(C|\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_t)$$

where C is a sequence of intent and I^t is the set of all possible sequences of intent with length t.

To find this sequence of intent C^*, MEMM keeps a variable $\alpha_t(i)$ which is defined to be the maximum probability of seeing a particular sequence of intent ending at intent i ($i \in I$) for question Q_t, given the observation sequence for questions Q_1 up to Q_t:

$$\alpha_t(i) = \max_{c_1,...,c_{t-1}} P(c_1, \ldots, c_{t-1}, c_t = i|\mathbf{f}_1, \ldots, \mathbf{f}_t)$$

This variable can be calculated by a dynamic optimization procedure similar to the Viterbi algorithm in the Hidden Markov Model:

$$\alpha_t(i) = \max_j \alpha_{t-1}(j) \times P(c_t = i|c_{t-1} = j, \mathbf{f}_t)$$

where $P(c_t = i|c_{t-1} = j, \mathbf{f}_t)$ is estimated by the Maximum Entropy Model.

Table 3 shows the best results of identifying user intent based on the Maximum Entropy Model and MEMM using the leave-one-out approach.

The results have shown that both models did not work for the data collected from unstructured scenarios (i.e., the baseline accuracy for intent identification is 63.4%). For structured scenarios, in terms of the overall accuracy, both models performed significantly better than the baseline (i.e., 49.3%). The MEMM worked only slightly better than the MEM. Given our limited data, it is not conclusive whether the transitions between questions will help identify user intent in a basic interactive mode. However, we expect to see more influence from the transitions in fully conversational QA.

[3] http://mallet.cs.umass.edu/index.php/

[4] http://svmlight.joachims.org/

[5] http://www.cs.waikato.ac.nz/ml/weka/

Features	MEM			SVM			DTree		
	un	s	ent	un	s	ent	un	s	ent
Baseline	66.2	51.5	56.3	66.2	51.5	56.3	66.2	51.5	56.3
TM, SQC	50.0	57.4	54.9	53.5	60.0	57.8	53.5	55.9	55.1
NEM, SQC	37.3	74.4	61.7	37.3	74.4	61.7	37.3	74.4	61.7
TM, SQ	61.3	64.8	63.6	57.0	64.1	61.7	59.9	64.4	62.9
NEM, SQC, SAC	40.8	**76.7**	64.3	38.0	74.4	61.9	49.3	**77.8**	68.0
TM, SQ, SAC	59.2	67.4	64.6	61.3	66.3	64.6	62.7	65.6	64.6
TM, NEM, SQC	54.2	75.2	68.0	54.2	**75.2**	68.0	53.5	74.4	67.2
TM, SQ, SA	63.4	71.9	68.9	58.5	71.5	67.0	67.6	75.6	72.8
TM, NEM, SQC, SAC	54.9	75.6	68.4	54.2	**75.2**	68.0	55.6	74.4	68.0

* un - unstructured, s - structured, ent - entire

Table 2: Performance of automatic identification of problematic situations

		MEM		MEMM	
		un	s	un	s
CONTINUE	P	64.4	69.7	67.3	70.8
	R	96.7	85.8	80.0	88.8
	F	77.3	76.8	73.1	78.7
RE-TRY /NEGOTIATE	P	28.6	76.2	37.1	79.0
	R	8.7	74.1	56.5	73.1
	F	13.3	75.1	44.8	75.9
SWITCH	P	-	-	-	50.0
	R	0	0	0	3.6
	F	-	-	-	6.7
Overall accuracy		62.7	72.2	59.9	73.7

* un - unstructured, s - structured

Table 3: Performance of automatic identification of user intent

5 Implications of Problematic Situations and User Intent

Automated identification of problematic situations and user intent have potential implications in the design of conversational QA systems. Identification of problematic situations can be considered as implicit feedback. The system can use this feedback to improve its answer retrieval performance and proactively adapt its strategy to cope with problematic situations. One might think that an alternative way is to explicitly ask users for feedback. However, this explicit approach will defeat the purpose of intelligent conversational systems. Soliciting feedback after each question not only will frustrate users and lengthen the interaction, but also will interrupt the flow of user thoughts and conversation. Therefore, our focus here is to investigate the more challenging end of implicit feedback. In practice, the explicit feedback and implicit feedback should be intelligently combined. For example, if the confidence for automatically identifying a problematic situation or an error-free situation is low, then perhaps explicit feedback can be solicited.

Automatic identification of user intent also has important implications in building intelligent conversational QA systems. For example, if *Continue* is identified during interaction, then the system can automatically collect the question answer pairs for potential future use. If *Switch* is identified, the system may put aside the question that has not been correctly answered and proactively come back to that question later after more information is gathered. If *Re-try* is identified, the system may avoid repeating the same answer and at the same time may take the initiative to guide users on how to rephrase a question. If *Negotiate* is identified, the system may want to investigate the user's particular needs that may be different from the general needs. Overall, different strategies can be developed to address problematic situations and different intents. We will investigate these strategies in our future work.

This paper reports our initial effort in investigating interactive QA from a conversational point of view. The current investigation has several simplifications. First, our current work has focused on factoid questions where it is relatively easy to judge a problematic or error-free situation. However, as discussed in earlier work (Small et al., 2003), sometimes it is very hard to judge the truthfulness of an answer, especially for analytical questions. Therefore, our future work will examine the new implications of problematic situations and user intent for analytical questions. Second, our current investigation is based on a basic interactive mode. As mentioned earlier, once the QA systems become more intelligent and conversational, more varieties of user intent are anticipated. How to characterize and automatically identify more complex user intent under these different situations is another direction of our future work.

6 Conclusion

This paper presents our initial investigation on automatic identification of problematic situations and user intent in interactive QA. Our results have shown that, once users are motivated in finding specific information related to their information goals, user behavior and interaction context can help automatically identify problematic situations and user intent. Although our current investigation is based on the data collected from a controlled study, the same approaches can be applied during online processing as the question answering proceeds. The identified problematic situations and/or user intent will provide immediate feedback for a QA system to adjust its behavior and adapt better strategies to cope with different situations. This is an important step toward intelligent conversational question answering.

References

Marco De Boni and Suresh Manandhar. 2003. An analysis of clarification dialogues for question answering. In *Proceedings of HLT-NAACL 2003*, pages 48–55.

John Burger, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, Sanda Harabagiu, David Israel, Christian Jacquemin, Chin-Yew Lin, Steve Maiorano, George Miller, Dan Moldovan, Bill Ogden, John Prager, Ellen Riloff, Amit Singhal, Rohini Shrihari, Tomek Strzalkowski, Ellen Voorhees, and Ralph Weishedel. 2001. Issues, tasks and program structures to roadmap research in question & answering. In *NIST Roadmap Document*.

Sanda Harabagiu, Andrew Hickl, John Lehmann, and Dan Moldovan. 2005. Experiments with interactive question-answering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 205–214, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Julia Hirschberg, Diane J. Litman, and Marc Swerts. 2001. Identifying user corrections automatically in spoken dialogue systems. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'01)*.

Gina-Anne Levow. 1998. Characterizeing and recognizing spoken corrections in human-computer dialogue. In *Proceedings of the 36th Annual Meeting of the Association of Computational Linguistics (COLING/ACL-98)*, pages 736–742.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, Madison, Wisconsin, July.

Diane J. Litman and Shimei Pan. 2000. Predicting and adapting to poor speech recognition in a spoken dialogue system. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 722–728.

Diane J. Litman, Marilyn A. Walker, and Michael S. Kearns. 1999. Automatic detection of poor speech recognition at the dialogue level. In *Proceedings of the 37th Annual meeting of the Association of Computational Linguistics (ACL-99)*, pages 309–316.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of Internatioanl Conference on Machine Learning (ICML 2000)*, pages 591–598.

Sharon Small, Ting Liu, Nobuyuki Shimizu, and Tomek Strzalkowski. 2003. HITIQA: An interactive question answering system: A preliminary report. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. In *Computational Linguistics*, volume 26.

Ellen Voorhees. 2001. Overview of TREC 2001 question answering track. In *Proceedings of TREC*.

Ellen Voorhees. 2004. Overview of TREC 2004. In *Proceedings of TREC*.

Marilyn Walker, Irene Langkilde-Geary, Helen Wright Hastie, Jerry Wright, and Allen Gorin. 2002. Automatically training a problematic dialogue predictor for the HMIHY spoken dialog system. In *Journal of Artificial Intelligence Research*.