Automated Performance Assessment in Interactive QA

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

ABSTRACT

In interactive question answering (QA), users and systems take turns to ask questions and provide answers. In such an interactive setting, user questions largely depend on the answers provided by the system. One question is whether user follow-up questions can provide feedback for the system to automatically assess its performance (e.g.,assess whether a correct answer is delivered). This self-awareness can make QA systems more intelligent for information seeking, for example, by adapting better strategies to cope with problematic situations. Therefore, this paper describes our initial investigation in addressing this problem. Our results indicate that interaction context can provide useful cues for automated performance assessment in interactive QA.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Search Process

General Terms: Experimentation, Performance

Keywords: Performance Assessment, User Behavior, Interactive Question Answering

1. INTRODUCTION

Interactive question answering has been identified as one of the important directions in QA research [1]. In interactive QA, users and systems take turns to ask question and provide answers. In such an environment, questions formed by a user not only depend on his/her information goals, but are also influenced by the answers from the system. Because of this dependency, our assumption is that user follow-up questions can provide feedback for the system to assess the status of preceding answers (e.g., whether a correct answer is delivered). The awareness of its own performance will enable the system to automatically adapt better strategies to cope with problematic situations. To our knowledge, there has not been much work that addresses this important aspect of interactive QA. This paper describes our initial investigation on this problem. Given a question Q_i and its corresponding answer A_i, the specific question examined here is whether the user language behavior in the follow-up question Q_{i+1} and the interaction context can help the system to assess its performance at answering the preceding question (A_i).

To address this question, we conducted a user study where users interacted with a *controlled* QA system to find information of interest. Our studies indicate that when the system fails to deliver a desired answer, users do exhibit some language behavior (e.g., rephrase of the question) in the follow-up question to respond to this problematic situation. User behavior and interaction context can provide important cues for a QA system to automatically identify problematic situations. Based on the data collected from our studies, we experimented with three classifiers (Support Vector Machine, Maximum Entropy Model, and Decision Tree). Our results indicate that the Decision Tree model can detect problematic situations with 73.8% accuracy, which is significantly better than the baseline.

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.
ACM 1-59593-369-7/06/0008.

2. USER STUDIES

To investigate the role of interaction context in automated performance assessment, we conducted a controlled user study where a human wizard was involved 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), it is 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. As a result, this simulation generated 56% error-free situations and 44% problematic situations. 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.

Eleven users participated in our study. Each user was asked to interact with our system to complete information

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

seeking tasks related to four specific scenarios. Each of the four scenarios was focused around a separate topic: *the 2004 presidential debates*, *Tom Cruise*, *Hawaii*, and *Pompeii*. As a result of this study, a total of 456 QA exchanges from 44 interactive sessions were collected, where each answer was annotated with a binary tag to indicate whether or not the answer was problematic.

3. PERFORMANCE ASSESSMENT

We formulate automated performance assessment as a classification problem. Given a question Q_i with a corresponding answer A_i, our goal is to decide whether A_i is problematic based on the follow up question Q_{i+1} and the interaction context. More specifically, the following set of features are used: (1) *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 show that the repetition of target type may indicate a question rephrase, which could signal a problematic situation has just occurred. (2) *Named entity matching (NEM)*: a binary feature indicating whether all the named entities in Q_{i+1} also appear in the 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. (3) *Similarity between questions (SQ)*: a numeric feature measuring the similarity between Q_{i+1} and Q_i. (4) *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. (5) *Similarity between Q_i and A_i(SA).* (6) *Similarity between Q_i and A_i only based on the content words (excluding named entities)(SAC).*

To measure the similarity between two chunks of text T_1 and T_2, we applied the following equation proposed by Lin [2]:

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

where $P(w)$ was calculated based on 1806 *pseudo documents* (i.e., question/answer pairs) from previous TREC evaluations.

We experimented with three classification approaches (Maximum Entropy Model from MALLET[2], SVM from SVM-Light[3], and Decision Trees from WEKA[4]) based on ten fold cross-validation (90% of data was used as training data and 10% as testing data in each trial). Table 1 shows the accuracy of the three approaches on identifying problematic/error-free situations using different combinations of features. The baseline was obtained by simply assigning the most frequently occurring class (i.e., 56% of correct situations in our data). The best performance for each model is highlighted in bold in Table 1. The Decision Tree model achieves the best performance of 73.8% in identifying problematic situations, which is more than 17% better than the baseline performance. Different combinations of features result in different performance in all three models. In general, the feature set that considers different forms of question/answer similarity works better than those that do not consider these aspects.

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

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

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

	Features	SVM	MaxEnt	DTree
(1)	Baseline	56.4	56.4	56.4
(2)	NEM, SQC	61.7	61.7	61.7
(3)	TM, SQ	63.7	64.1	61.7
(4)	TM, SQ, SA	**68.3**	**69.2**	72.1
(5)	TM, NEM, SQ, SQC, SA	66.8	66.6	71.3
(6)	TM, NEM, SQ, SQC, SA, SAC	67.3	67.7	**73.8**

Table 1: Accuracy of automated performance assessment based on three approaches

Identification of problematic situations can be considered as implicit feedback. One might think that an alternative way is to explicitly ask users for feedback (for example, with a feedback button). However, soliciting feedback after each question not only will frustrate users and lengthen the interaction, but also it may not be possible for certain devices (e.g., PDA). Therefore, our focus here is to investigate the more challenging end of identifying problematic situations through implicit feedback. In real interaction, explicit and implicit feedback should be intelligently combined. For example, if the confidence for identifying a problematic situation or an error-free situation is low, then perhaps explicit feedback can be solicited.

4. CONCLUSION

This paper presents our initial investigation on automated performance assessment in interactive question answering. Our studies indicate that when a problematic situation occurs (i.e., retrieved answer does not appear to be correct), users exhibit distinctive behavior such as rephrasing the question. Follow-up questions and interaction context can provide useful cues for the system to automatically evaluate its performance. Although our current evaluation is based on the data collected from our study, the same approaches can be applied during online processing as the question answering session proceeds. Such performance assessment can provide feedback directly to a QA system as to what questions the system may have correctly answered and what questions the system may have trouble with. This will not only allow the system to automatically adapt better strategies during online processing but also provide a mechanism to automatically build databases of question answering pairs for other applications(e.g., collaborative question answering).

5. REFERENCES

[1] J. Burger and et al. Issues, tasks and program structures to roadmap research in question & answering. In *NIST Roadmap Document*, 2001.

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