Michigan State University at the 2007 TREC ciQA Evaluation

Chen Zhang Matthew Gerber Tyler Baldwin
Steve Emelander Joyce Y. Chai Rong Jin

Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824, USA
{zhangch6, gerberm2, baldwi96, emeland4, jchai, rongjin}@cse.msu.edu

1 Introduction

This is our first participation in the ciQA task. Instead of exploring conversation strategies in question answering [3, 4], we decided to focus on simple interaction strategies using relevance feedback. In our view, the ciQA task is not designed to evaluate user initiative interaction strategies. Since NIST assessors act as users, the motivation to take an initiative is lacking. It is not clear how to encourage the assessors to take the initiative (e.g., by asking additional questions) during the interaction process. We feel in such a setting, relevance feedback or any kind of system initiative interaction strategies seem more appropriate. Therefore, we have focused on variations of relevance feedback in this year's evaluation.

For the initial runs, our two submissions were based on two distinct approaches, a heuristic approach and a machine learning approach. Since only two interactive runs can be submitted for evaluation, we decided to focus on one aspect of variation. The only difference between the two interactive and final run systems is how the feedback is solicited and incorporated. Since there are many parameters inherent in the evaluation that affect the outcome of the final runs, only varying one parameter will hopefully allow us to make some preliminary observations about how feedback solicitation can affect final performance.

Although manual runs were allowed in this evaluation, all of our runs were created automatically. The following steps were taken during the evaluation. For each topic, the system first generated a query based on its question template and narrative and used this query to retrieve relevant documents. The retrieved documents were then segmented into sentences, which were further ranked and put together as the initial run results. The interactive web pages were generated based on the results from the initial runs. These pages were accessed by NIST assessors. Feedback from assessors was used to create the final run results.

In the following sections, we describe in detail the steps taken to create our initial runs and final runs. We also discuss what we have learned from this exercise.

2 Initial Runs

2.1 Query Formulation and Document Retrieval

The system applied a template dependent strategy to formulate queries for each topic. For most templates, the phrases in every slot of the templates and named entities in the narratives were used to form queries. The named entities were extracted using BBN's named entity identifier [2]. For the template *What [relationship] exist between [entity] and [entity]?*, the filler of the *[relationship]* slot (e.g., *financial relationships*, *common interests*) was not used in query formulation. For the *financial relationship* topics, finance-related terms such as *money*, *trade*, etc. from the narrative were used in the query. Figure 1 shows an example of query formulation. Please note that our query has the form of phrase-and-phrase structure, which was utilized in the latter procedure of answer extraction and ranking. When these queries were used to retrieve documents, they were treated as bags-of-words.

Template: What evidence is there for transport of [**military equipment and weaponry**] from [**South Africa**] to [**Pakistan**]?

Narrative: The analyst is interested in **South African** arms support to **Pakistan** and the effect such support or sales has on relations of both countries with **India**. Additionally, the analyst would like to know what nuclear arms involvement, if any, exists between **South Africa** and **Pakistan**.

Query: military equip weaponry, South Africa, Pakistan, India

Figure 1: An example of query formulation

The Lemur[1] retrieval engine was used to retrieve the top 100 most relevant documents from the AQUAINT II corpus. The retrieved documents were further combined with 100 documents (per topic) provided by NIST, which resulted in an average of 155 documents per topic. During this combination, a simple algorithm was applied to ensure the retrieval score for each document was on the same scale despite the fact that documents were coming from two different sources. These documents were used for answer extraction and ranking.

2.2 Answer Extraction and Ranking

The documents produced in the previous step were split using the sentence detector provided by the OpenNLP toolkit[2]. The initial run results were produced by ranking these sentences and selecting the top 7000 characters as answers. Two different ranking algorithms were used to produce two initial run submissions.

2.2.1 Heuristic Approach

This approach was based on the following heurisitcs developed at the University of Waterloo [10], which exhibited the best performance in the 2006 initial run evaluations [5]:

1. Rank sentences by the number of query phrases they contain. A sentence is judged to contain a phrase if it contains any term in the phrase.
2. Break ties by the number of query terms the sentences contain.
3. Break further ties by the average inverse document frequency (IDF) for all non-stop words in each sentence.

Since, for some templates (e.g., *What evidence is there for transport of [goods] from [entity] to [entity]?* and *What financial relationships exist between [entity] and [entity]?*), special named entities signal whether a sentence contains a quality answer, we extended the original heuristic by incorporting a named entity check as follows:

1. Rank sentences by the number of query phrases they contain.
2. Break ties by the number of query terms they contain.
3. Break further ties by whether they contain special named entities.
4. Break further ties by the average IDF.

Named entities were extracted using BBN's IdentiFinder. Different templates required checking for different types of named entities. For example, for the transportation template (*What evidence is there for transport of [goods] from [entity] to [entity]?*), the named entity types *quantity*, *location*, and *cardinal* were used. The extended heuristics have shown to improve the results on 2006 data compared to the original heuristics (see Section 2.2.3).

[1] http://www.lemurproject.org
[2] http://www.opennlp.org

2.2.2 Machine Learning Approach

Based on the publicly available data from the 2006 ciQA evaluation [5], we developed a machine learning approach to predict whether a sentence is considered a good answer to a complex question. We examined the candidate sentences for the 30 topics from 2006 and labeled those that were considered quality answers, i.e., those that cover the meaning of the answer nuggets.

We formulated the machine learning approach as a binary classification problem, and used a logistic regression model [7] to perform the classification. For each sentence, the classifier predicted the probability that it belonged to the positive class (true positive answers being quality answers). The probabilities were then used to rank the candidate answer sentences.

We identified four groups of features for the classification:

- Query-independent sentence-level features:
  - Sentence length, in terms of non-whitespace characters.
  - The average IDF of non-stop words in the sentence.
  - The average log-IDF of non-stop words in the sentence.
- Query-dependent sentence-level features:
  - The retrieval score of the document containing the sentence.
  - The number of query phrases the sentence contains. A sentence is judged to contain a phrase if it contains any term in the phrase.
  - The number of query terms contained in the sentence.
- Document-dependent sentence-level features:
  - The percentage of sentences in the document that share terms with the sentence under consideration, which is the same as *lexical bonds* in [10].
  - The position of the sentence in the document.
  - The position of the sentence in the paragraph containing it.
- Binary features indicating whether or not the sentence contains certain types of named entities:
  - Countries, cities, states, provinces
  - Physical locations (bodies of water, mountains, continents, etc.)
  - Organization names
  - Person names
  - Substance names
  - Dates
  - Cardinal numbers
  - Money amounts
  - Percentages
  - Quantity numbers

We trained a model for each of the template types seperately, except for the *common interests* and *familial/organizational ties* templates. For the latter two templates, a model trained on the entire dataset was used.

2.2.3 Empirical Comparison of the Two Answer Ranking Approaches

Before we applied these approaches on this year's data, we evaluated these two approaches using data from the 2006 evaluation. For the machine learning approach, we used leave-one-topic-out cross validation. The evaluation metric was the official pyramid F-score measurement [9, 8]. The results are shown in Table 1.

The results demonstrate that both approaches outperformed last year's automatic run results. Additionally, the machine learning approach exhibits a small advantage over the simple heuristic approach.

Table 1: Empirical results from two sentence ranking approaches on 2006 ciQA data

	Precision	Recall	F ($\beta = 3$)
Heuristic	0.080	0.441	0.282
Learning	0.085	0.450	0.293

3 Interaction

The interaction component plays a vital role in the ciQA evaluations. One observation about the current evaluation is that results are judged after the interaction is completed rather than during the interaction. Intermediate results after each run of interaction are not considered. Thus, it does not matter if a system is fast or slow in delivering its final answer. Based on this observation, we designed two types of interactive systems. In one system, after each round of interaction, user feedback is immediately passed to the backend processing components (e.g., answer extraction and re-ranking) to generate a new set of candidate answers. We call this sytem the *interaction system*. In the second system, during interaction, the system only collects feedback from the user. Once the interaction is completed, the feedback is sent to the backend to generate potential answers. We call this system the *collection system*. The reason we started with relevance feedback as the basic interaction strategy is two-part. First, relevance feedback has shown to be effective in information retrieval [6]. Second, relevance feedback provides a reasonable baseline for our future work on other interaction strategies.

3.1 The Interaction System

A traditional relevance feedback approach is used in the interaction system. For each interaction round, the top 20 sentences are displayed to the user for relevance feedback. Based on the feedback, the system re-ranks the remaining sentences in the list and shows the next top 20 on the re-ranked list.

The re-ranking function is a variation of the Rocchio algorithm [1]. After each interaction, the system computes a weight for each term in the vocabulary based on all feedback given:

$$w(t) = count(t, q) + \frac{1}{|pos|} \sum_{f \in pos} count(t, f) - \frac{1}{|neg|} \sum_{f \in neg} count(t, f) \quad (1)$$

where q is the query used for document retrieval and answer extraction, pos is the set of positive feedback answers, neg is the set of negative feedback answers, and $count(t, f)$ is the frequency of term t in feedback answer f.

Our modification to this function ignores terms with negative weights:

$$w'(t) = \begin{cases} w(t) & w(t) \geq 0 \\ 0 & w(t) < 0 \end{cases} \quad (2)$$

This is based on the assumption that for this specific ciQA task, positive answers are much fewer than negative answers, so positive feedback should carry much more weight than a negative feedback.

Terms and their corresponding weights become a new, modified query. The system then re-ranks the sentences based on their similarities with the modified query.

3.2 The Collection System

The collection system is designed to allow users to browse through sentences to provide relevance feedback with the help of a list of automatically generated filters. Our assumption is that many users are knowledgable. They may want to control the types of sentences they see and provide feedback. The collection system does not undertake any backend processing, but rather collects as much relevant information as possible. Figure 2 shows a snapshot of the collection system. The left panel contains the filters generated by the system. Each filter is either a query term or an informative non-query noun that appears in the question template or narrative and has hyponyms or meronyms in top 200 ranked sentences. The filter list also contains salient named entities from the answer passages. The user can select one or more filters that are related to his or

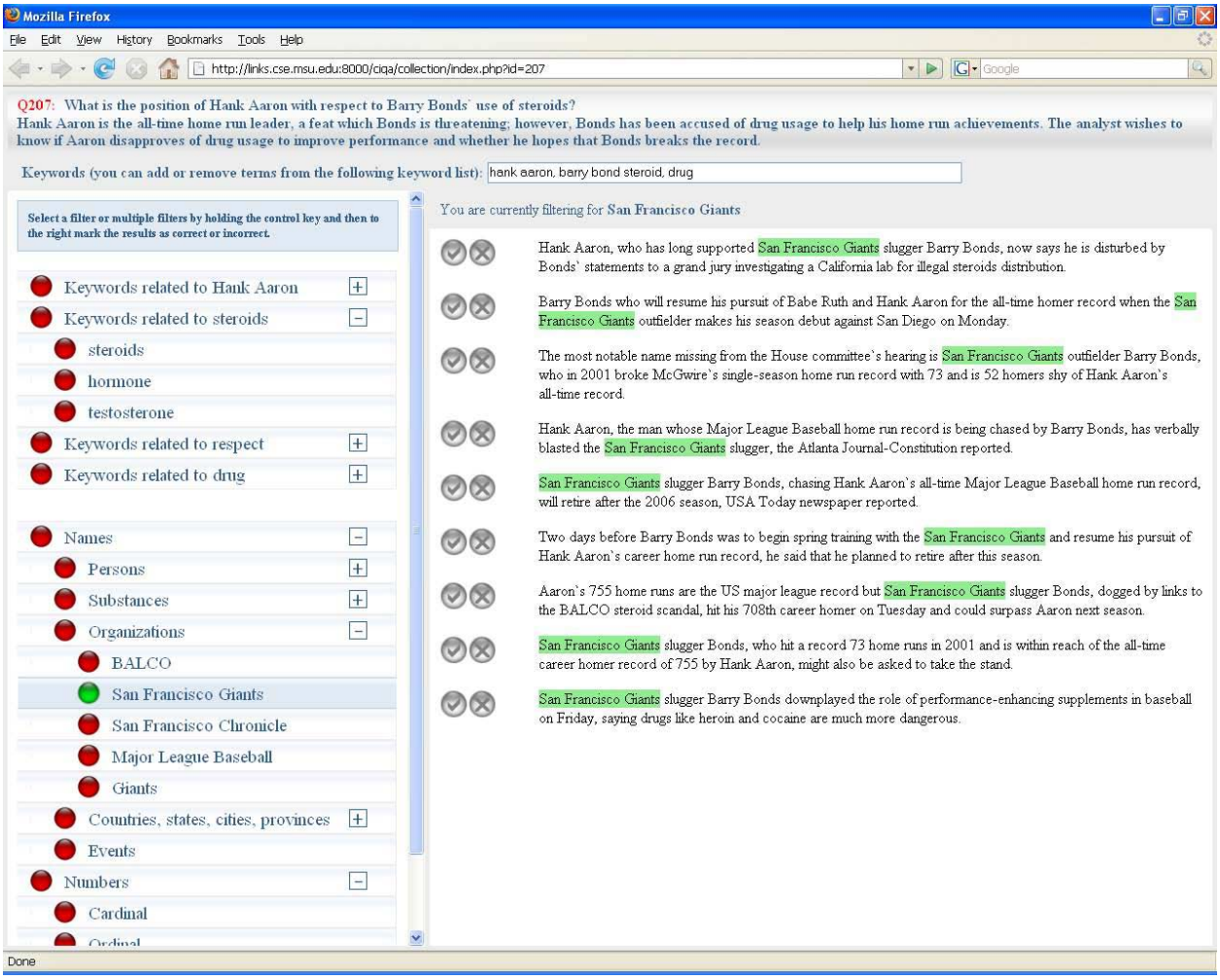

Figure 2: A snapshot of the collection system

Table 2: Evaluation results of two initial runs on 2007 data

Run tag	Sentence ranking approach	Average F score
MSUciQAiHeu	Heuristic	0.359
MSUciQAiLrn	Learning	0.387

Table 3: Evaluation results of two final runs on 2007 data

Run tag	Interface	Average F score	Improvement over baseline
MSUciQAfInt	*Interaction*	0.370	0.011
MSUciQAfCol	*Collection*	0.361	0.002

her information needs. When the filters are selected, the sentences in the right panel are updated to reflect the filter selection. Users can then give feedback on the displayed sentences.

After collecting the feedback from the user, the final sentence list is produced using the same re-ranking method as the interaction system.

4 Evaluation Results

We submitted two initial runs and two final runs. We picked the initial run results generated by the heuristic approach to solicit user feedback and create our final run results.

4.1 Initial Run Results

The evaluation results of our two initial runs are shown in Table 2. The heurisitic approach achieved an F-measure score of 35.9%, and the learning-based approach achieved an F-measure score of 38.7%. These results are ranked third and sixth among eleven automatic initial runs.

4.2 Final Run Results

The initial run results based on the heurisitic approach were used in the interaction systems for feedback. Unfortunately, a major network failure occured during the two days when the NIST assessors tried to interact with our pages. On the first day, the entire MSU network was unaccessible. Based on the limited feedback we did receive, two of our final runs (one from each interaction system) were produced. Table 3 shows the results of our final runs. Compared to the baseline system (e.g., MSUciQAiHeu), there was a small improvement for both final runs.

5 Discussion and Conclusions

Although the experiments with the interactive systems did not go smoothly due to network issues, we were still able to make some observations based on the limited feedback our systems received.

First, as shown in Table 4, the amount of feedback received by the collection system is significantly less than the amount received by the interaction system. Note that, despite the network problems, since the two interaction systems for a particular topic were accessed at roughly the same time, it is possible to compare the two systems. Our original assumption was that because no backend processing was involved in the collection system, all the interaction time (i.e., 5 minutes) would be devoted to collecting feedback. Thus more feedback should be received compared to the interaction system where a portion of the interaction time would be used for the backend processing. However, as shown in Table 4, the results disconfirmed our assumption. The collection interface requires a user to be more motivated to choose different combinations of filters to navigate through results. Possible lack of motivation and the more complex interface may contribute to the ineffectiveness of the collection system.

Table 5 shows the percentage of positive and negative feedback received for both interfaces. In general, the ratio of positive feedback is quite high, which means our initial sentence ranking algorithm provides a

Table 4: Per-topic feedback statistics of user relevance feedback for our two interfaces

	Min	Max	Average
Interaction	7	105	30
Collection	2	80	14

Table 5: Statistics of positive and negative feedbacks for our two interfaces

	+	-	No feedback
*Interaction**	60%	31%	9%
Collection	62%	15%	23%

*Statistics on the interaction interface only count the top 10 sentences for each topic

reasonable baseline. This ratio is even higher for the collection interface. Since positive feedback is much more valuable than negative feedback, the idea of enabling the user to explore their desired information through filters (as in the collection system) appears to have some merit.

Furthermore, we are interested in examining the consistency of the feedback. There were 247 sentences (for all topics) for which both interfaces received user feedback. Out of these sentences, 25 of them were provided with conflicting feedback by the same assessor. Figure 3 shows the distribution of these conflicts among the eight assessors. We can see that the feedback consistency is largely dependent on individual assessors (e.g., for one assessor, more than one third of the feedback was conflicting). This observation indicates that it is sometimes difficult for a human to judge the correctness of an answer, which motivates further development of evaluation methodologies for the ciQA task.

6 Acknowledgements

We would like to thank BBN for the IdentiFinder software. This work was partially supported by the Disruptive Technology Office (DTO) Phase III program for the Advanced Question and Answering for Intelligence (AQUAINT).

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[2] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *The Proceedings of Fifth Conference on Applied Natural Language Processing*, pages 194–201, 1997.

[3] J. Chai, T. Baldwin, and C. Zhang. Automated performance assessment in interactive question answering. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, 2006.

[4] J. Chai, C. Zhang, and T. Baldwin. Towards conversational qa: Automated identification of problematic situations and user intent. In *Proceedings of the International Conference on Computational Linguistics/Association for Computational Linguistics*, 2006.

[5] H. T. Dang, J. Lin, and D. Kelly. Overview of the trec 2006 question answering track. In *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.

[6] J. Koenemann and N. J. Belkin. A case for interaction: a study of interactive information retrieval behavior and effectiveness. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 205–212, New York, NY, USA, 1996. ACM.

[7] S. le Cessie and J. C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

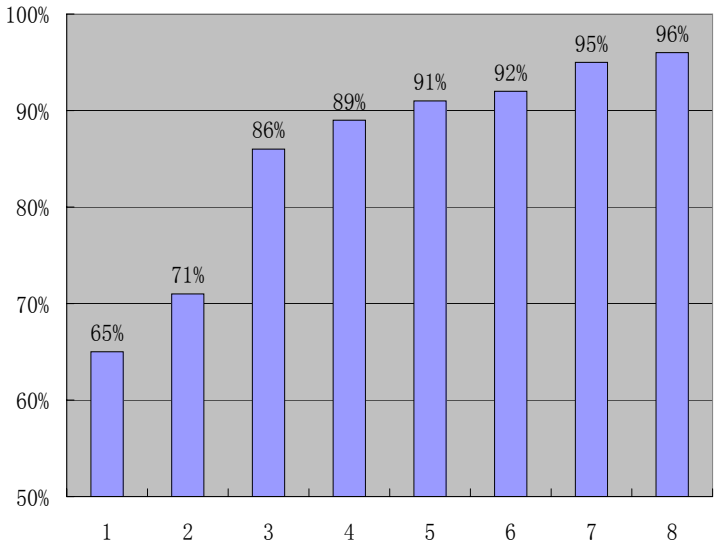

Figure 3: Consistency of feedbacks given to both interfaces for eight assessors, ranking from the least consistent to the most consistent

[8] J. Lin. Is question answering better than information retrieval? towards a task-based evaluation framework for question series. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 212–219, Rochester, New York, April 2007. Association for Computational Linguistics.

[9] J. Lin and D. Demner-Fushman. Will pyramids built of nuggets topple over? In *Proceedings of the 2006 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2006)*, pages 383–390, New York, New York, 2006.

[10] O. Vechtomova and M. Karamuftuoglu. Identifying relationships between entities in text for complex interactive question answering task. In *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.