Fora: Leveraging the Power of Internet Communities for Question Answering

Gu Xu, Hang Li, Wei-Ying Ma
Microsoft Research Asia
4/F, Sigma Center, No.49, Zhichun Road
Haidian, Beijing (100080), China
{guxu, hangli, wyma}@microsoft.com

ABSTRACT

This paper introduces a system for searching question answer pairs automatically extracted from the discussions in internet communities. The system, named Fora, aggregates discussions from multiple forums and newsgroups in the same domain, automatically extracts question answer pairs from the data, and provides searches of the question answer pairs. The system also offers expert search, query suggestion, page search, and other features. This paper explains the main features and technologies of Fora. It describes how the system extracts and ranks question answer pairs.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval; H.4.0 [**INFORMATION SYSTEMS APPLICATIONS**]: General

General Terms

Algorithms, Experimentation

Keywords

Internet Community, Question Answering

1. INTRODUCTION

Internet communities, including web forums, newsgroups, and mailing lists, provide shared spaces for open communications and discussions on commonly interested topics. Users can post their questions to the discussion groups of the communities and get answers from the others. The amount of knowledge in the form of QA accumulated in various communities is tremendous, covering all sorts of topics and growing rapidly. Unfortunately, such valuable knowledge is not fully leveraged in practice. Similar questions are repeatedly asked and answered at different places all the time.

A recent but related scheme is Community Question Answering (CQA), which is a type of forums specialized for question answering, e.g. [3]. CQA offers an environment for users to question and answer. It becomes a successfully model and starts to accumulate a large amount of valuable QA knowledge as well. However, how to effectively utilize the data and also communities is still an open problem.

Web search engines have certain capability of searching forums and newsgroups. However, they usually only index a small portion of pages from web forums and newsgroup archives, and do not consider dealing with the specialties of QAs in their searches. They return search results at page level and usually it is hard to find the answers in a long discussion thread. Some search systems, like Google Groups [1] and Omgili [2], organize the discussions in a better way. However, they still cannot help users to directly find the answers.

Question answering is a powerful means for information retrieval, when compared with traditional keyword search. In question answering, users type natural language questions instead of keywords to look for the information they want. Existing work mainly focused on answering simple questions such as factoid questions ("when was Windows Vista released", or "what is xbox") [5, 8], list and definition questions [9, 10], and other types of questions [11]. FAQ Finder [5] is an early QA system using Frequently Asked Questions (FAQ) files from USENET newsgroups. However these FAQ files, containing clean question and answer pairs, are not commonly available especially for domain-specific questions. All of them cannot effectively handle complex questions, for example, "are there any good games on Xbox 360".

In this paper, we present a new search system named *Fora* (the plural form of "forum" in Latin), which aims to solve the problems described above, that is, leveraging the rich QA knowledge in the discussions in the internet communities and providing searches of question answer pairs. Fora automatically extracts question answer pairs existing in forums and newsgroups, and provides search services of the extracted question answer pairs. Users can type keywords or even natural language questions to retrieve the question answer pairs. This can help users to quickly reach the answers to their questions and drastically reduces the workload in their QA searches. Fora also provides expert search, query suggestion features to further enhance the search experiences of users.

2. TECHNICAL CHALLENGES

Several technical challenges stand in the way of developing a QA system.

QAWeb 2008, April 22, 2008, Beijing, China.

The questions and answers in forums exist in *discussion threads*. Usually, a thread contains multiple posts, the post at the beginning consists of the original question, and several posts at the end consist of answers to the question. Since the format and style of discussion is completely free to the users, automatically finding the question and the answers from the thread becomes challenging. Obviously not all the replies are answers, for example, posts with sentences of "I also want to know the answer" and "thanks, it works" are not true answers, so are the posts having sentences of "I do not know" and "please search it". Sometimes, multiple answers exist, but their qualities may also vary. For example, the single-word answer "yes" should not be as good as a detailed answer. It appears necessary to rank the likely answers according to the quality of them.

Another challenge arises when users search the extracted question answer pairs. There might always be mismatch between the question or query submitted by the user and the questions existing in the question answer pairs. This is because for any single issue there might be many ways to describe it in natural language. If the question is "are there any good games on Xbox 360", then the question "I want to find games for Xbox 360" shares the same meaning. New terms, technical terms, synonymous expressions, idiomatic expressions, abbreviations, incomplete sentences, typos, grammatical errors are very common in the discussions and the queries, and all of them can increase the possibility of the mismatching. Therefore, a sophisticated QA system must have effective means to deal with the problem.

Automatic QA extraction cannot be sufficient. When the system cannot find a relevant QA pair, it is helpful if it can provide a list of active people in the communities who might be able to answer the question. In the internet communities, not only the data but also the people are precious assets. How to effectively leverage the human asset is another issue one must consider.

Yet another technical challenge lies in the crawling of forum data. To collect data from different forum sites, ideally one wants to use a single crawler which can handle data in different formats from different sites. The link structures in different forum sites may differ largely, and there might also be many duplicated web pages. Efficient and effective crawling of forums is not an easy issue as it appears to be. Incremental crawling is another issue one must address, as the data in forums change dynamically.

3. OVERVIEW OF SYSTEM

Fora consists of four major components. Fig 1 shows the overview of the system.

Crawler

Crawler is responsible for fetching the data from various forums and newsgroups, and converting them into a unified format for ease of processing. Discussions from different sources usually need different ways to download and parse. The crawler in Fora is generic and can handle the specialities existing in conventional forum and newsgroup sites. The crawler is also able to conduct de-duplication of crawled pages and automatically learn the URL patterns of forum sites.

Extractor

Extractor is the core component of Fora. It filters out non-answer posts and sorts answers according to their quality

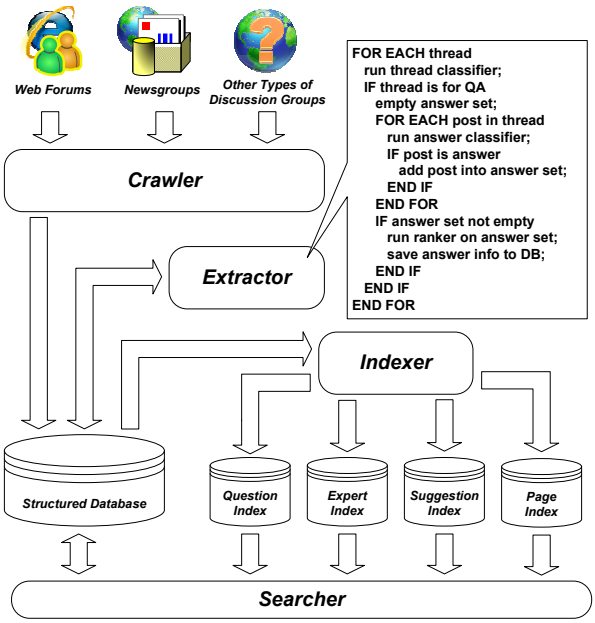

Figure 1: Fora System Overview

(likelihood of being good answers). The details of it are depicted in Fig 1. Given a thread, extractor automatically identifies the question and answers from the thread. The details of QA extraction and ranking are described in Section 5.

Indexer

Indexer creates multiple indexes for search. Question index is for finding similar questions; expert index is for searching experts; suggestion index is for retrieving suggestions to queries; and page index is for traditional keyword search on pages.

Searcher

The searcher provides search features in front end, as explained in Section 4. How to efficiently match user queries to the questions in the index is a commonly interested issue in QA research [5, 4]. Due to the difficulties aforementioned, term matching techniques, e.g. BM2500 [7], combining with term expansion are employed in Fora system to retrieve similar questions. It is efficient and also effective [12] especially when the index volume becomes large, for example million of questions. Users often use almost the same sentences to describe a question.

The searcher also collects users' feedbacks. Users are allowed to provide feedbacks on the automatically extracted answers (give "thumb up" or "thumb down" on the ranked answers). In this way, Fora system can continuously improve the quality of the QA search.

4. SEARCH FEATURES

There are four major search (or QA) features at Fora.

QA Search

The major feature in search is QA pairs search. Given the query (or question), the system returns a number of QA pairs which might be relevant to the query. The QA pairs are from different forums and newsgroups, automati-

cally extracted in advance and stored in the QA database. For each question in a thread, there might be multiple likely answers associated, ranked according to their likelihood of being good answers to the question. (See Fig. 2)

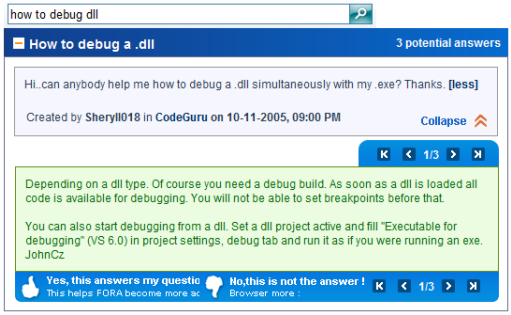

Figure 2: Fora QA Search

Expert Search

The system also returns a list of people who might be experts on the given topic in search. (See Fig. 3)

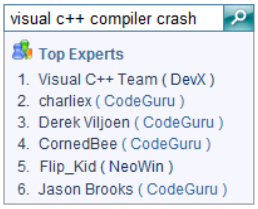

Figure 3: Fora Expert Search

Query Suggestion

When the user types a query, the system provides a number of suggested queries to help user better form his/her query. (See Fig. 4)

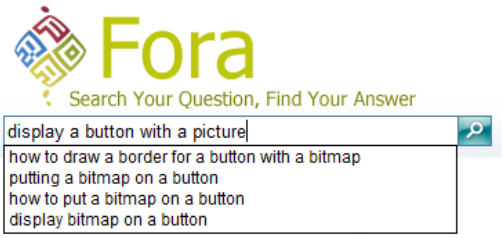

Figure 4: Fora Query Suggestion

Page Search

Users can also conduct page level search, and this is similar to traditional keyword search. (See Fig. 5)

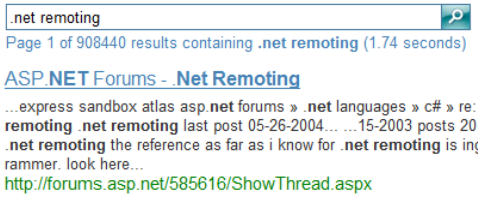

Figure 5: Fora Page Search

5. QUESTION ANSWERING

In Fora, the extractor employs the following method to automatically extract QA pairs, using heuristic rules and Ranking SVM [6].

First, it uses heuristic rules to filter out unlikely answers. The rules may look like "*a post cannot be an answer if it is less than 10 words, created by the question owner, and begins with the word 'thanks'*". Next, it uses Ranking SVM to score and sort potential answers. The Ranking SVM model utilizes a number of features and is trained with certain amount of labeled data in advance. Three types of features are used: content features, author features, and thread structure features. Content features make use of clues from texts, e.g. inclusion of indicative terms/phrases and word overlaps to its question. Author features capture the author's information, e.g. number of replies (activeness) which the author has submitted. Thread structure features indicate relative position of a post on the "thread tree", etc.

In search, the questions (or threads) are ranked according to their relevance to the query. Within each thread, the likely answers are ranked according to the their scores assigned by Ranking SVM.

We evaluated the performance of answer ranking. The Ranking SVM model was trained with a manually labeled dataset containing 1,169 threads from www.codeproject.com. These data are carefully selected to ensure the balance of answers and non-answers, and the number of posts in single thread. We also collected 63,582 threads from forums.microsoft.com as test data, and took the questioner's annotated "answers" as the ground truth. In these test data, 46.7% threads have no answer, only 18.5% posts are marked as "answer", and 3.3 posts per thread in average. The performance was measured in terms of P@1 and P@2.

Two baseline methods are considered. One is the simple heuristic of sorting the replies in a chronological order. It simulates the usual way of people's reading a discussion thread. It is also very effective because the most common pattern in discussions is that one user gives an answer and the questioner makes an acknowledgment at the end. The other baseline is to employ an SVM classifier and use the classification scores to rank the answers. The results are shown in Table 1. We can see that Ranking SVM significantly outperforms the two baseline methods.

Table 1: Performance on Question Answering

Algorithm	P@1	P@2
Heuristics	32.6%	31.6%
SVM Classifier	36.8%	36.0%
Ranking SVM	49.6%	44.1%

6. CURRENT STATUS

Fora crawled about 1.5 million threads from six forum sites on Microsoft technologies. The Fora system is running internally at Microsoft and MS employees can use it in their daily work. Many positive feedbacks to the system have also been obtained from the users and they say that Fora can really help them in finding QA information.

7. ACKNOWLEDGMENTS

We thank Ruochi Zhang, Matt Scott, Hao Xia, Bin Luo, and Erdong Chen for their contributions to the development of Fora system.

8. REFERENCES

[1] Google groups. *http://groups.google.com/*.

[2] Omgili forum search engine. *http://www.omgili.com/*.

[3] Yahoo! answers. *http://answers.yahoo.com/*.

[4] E. Agichtein, S. Lawrence, and L. Gravano. Learning search engine specific query transformations for question answering. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 169–178, New York, NY, USA, 2001. ACM.

[5] R. D. Burke, K. J. Hammond, V. A. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. Technical report, Chicago, IL, USA, 1997.

[6] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[7] S. Robertson. Overview of the okapi projects. *Journal of Documentation*, 53:3–7(5), 1 January 1997.

[8] E. M. Voorhees. The trec-8 question answering track report. In *Text REtrieval Conference*, 1999.

[9] E. M. Voorhees. Overview of the TREC 2001 question answering track. In *Text REtrieval Conference*, 2001.

[10] E. M. Voorhees. Overview of the TREC 2003 question answering track. In *Text REtrieval Conference*, 2003.

[11] E. M. Voorhees and H. T. Dang. Overview of the TREC 2005 question answering track. In *Text REtrieval Conference*, 2005.

[12] W. Xi, J. Lind, and E. Brill. Learning effective ranking functions for newsgroup search. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 394–401, New York, NY, USA, 2004. ACM.