Refinement of the ACP2P by Sharing User-Feedbacks and Learning Query-Responder-Agent-Relationships *

(Extended Abstract)

Tsunenori Mine
Kyushu University
744 Motooka, Nishiku
Fukuoka, Japan
mine@acm.org

Akihiro Kogo
Kyushu University

Satoshi Amamiya
ACN co. ltd.

Makoto Amamiya
Osaka
Institute of Technology
Osaka, Japan
amamiya@al.is.kyushu-u.ac.jp

ABSTRACT

This paper proposes two methods for improving the retrieval accuracy of the Agent-Community-based Peer-to-Peer information retrieval (ACP2P) method. One uses user feedbacks exchanged in a community. The other uses query-learning methods that make a middle agent to learn query-responder agent relationships. The latter methods are useful not only for improving the retrieval accuracy, but also for reducing communication loads. We conduct several experiments with test collections so that the evaluation can be done in an objective manner. The experimental results illustrate the validity of our proposed methods.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: relevance feedback, retrieval models, search process, selection process

General Terms

Algorithm, Design, Experimentation, Performance

Keywords

Multi Agent, Peer-to-Peer, Information Retrieval, ACP2P, User Feedback, Query Learning

1. INTRODUCTION

This paper discusses the methods for improving the ACP2P method[2]. First, we propose a new formula for calculating a document score to a query by using user feedbacks. The user feedbacks can be positive or negative judgement for a document to a query. Considering the user feedbacks to calculate the document score, highly evaluated documents will be given a higher score and ranked at a higher rank. The retrieval accuracy will accordingly be increased. In addition, we propose novel methods that improve the query-multicasting technique performed by a middle agent called a Portal Agent (PA). Using a history of query-responder agent relationships learned by the PA, the methods select Information Retrieval (IR) agents that would have documents highly relevant to a query and issue the query to the IR agents. The experimental results illustrate the validity of the methods.

*(Produces the copyright information for AAMAS 2009). For use with aamas2009.CLS and aamas2009-extendedabs.cls

Cite as: Refinement of the ACP2P by Sharing User-Feedbacks and Learning Query-Responder-Agent-Relationships, (Extended Abstract), Tsunenori Mine, Akihiro Kogo, Makoto Amamiya, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1341–1342

2. REFINEMENT OF ACP2P

Overview of ACP2P.

The ACP2P method[2] assigns an IR agent to each user. In order to search for information relevant to a query issued by a user, his/her IR agent mainly communicates with other IR agents in the same community as it belongs to. If the IR agent does not know their addresses, it can ask a PA in the community. The PA has a role of a manager or a router in the community, keeps the addresses of all the member IR agents in its community and helps to multicast a query to them. The PA is a representative of the community and also a member IR agent of the upper community. This can comprise a hierarchical community structure.

User Feedbacks and Query-Learning.

An IR agent obtains its user's relevancy judgements whether retrieved results returned by his/her IR agent are relevant or not, and adds them in its retrieved history. When searching for a relevant document, the IR agent calculates the similarity score between a query and a document by the following equation $Sim_f(Q, D) = Sim_d(Q, D) * r_{Q,D}(E)$. Where $Sim_d(Q, D)$ is the score of a document D to a query Q and calculated by a formula based on BM25[3]. E is a value of relevancy judgement given by a user and $r_{Q,D}(E)$ returns a positive number uniquely determined by E to a pair of Q and D. In section sec:simulation, $r_{Q,D}(E)$ returns 4, 3, 0.5, 0.5 when E takes highly relevant, relevant, partially relevant and irrelevant. When firstly issuing a query, a query-sender IR agent does not know to which IR agents it should issue the query. Thus, the query-sender IR agent uses a query-multicasting technique, gathers answers from all the member IR agents and makes a list of some specified number of agents in descending order according to the maximum document scores they returned. Since the query-multicasting technique has the problems that it needs huge communication loads and the same query tends to be issued to almost the same target agents, we propose two query learning methods: QL1 and QL2. The two methods use the Query-

Multicast-Request-History (Q/MRH) that consists of a list of four types of fields: $query$, $from$, $list-of-responder$ and $list-of-target$. Each type of the fields holds an issued query, the address of a query-sender IR agent, the list of responder agents to the query, and the list of target IR agents returned to the query-sender IR agent, respectively. The QL1 considers only the latest record in the Q/MRH. When being asked to do multicasting a query, a PA looks up the query from the Q/MRH. If there is the query in it, the IR agent in the $from$ field of the Q/MRH is determined as one of target query-receiver IR agents. IR agents, which are not in the $from$ and $list-of-target$ fields of the query, are selected by multicasting the query as target IR agents. The QL2 firstly makes a list of IR agents responding to each query. Then, from next time, it selects some specified number of target IR agents in the $list-of-responder$ in order and makes the PA multicast the query to them, with accumulating the previous retrieval results. The QL2 drastically reduces communication loads since the number of target IR agents receiving a query is limited except for the first multicasting time of each query. In addition, since the variety of target IR agents will be selected in order, the redundancy of retrieved documents will be decreased and greater number of relevant documents will be obtained.

3. EXPERIMENT AND DISCUSSION

We use two test collections, NTCIR3 WEB and NTCIR4 WEB[1], for the experiments. The document set of the collections consist of about 11 million Web documents. The 150 thousand documents among them are given five-level rating; highly relevant, relevant, partially relevant and irrelevant. Only the documents evaluated highly relevant or relevant to a query are dealt with as the ones relevant to the query in the experiments. The number of retrieval assignments given by NTCIR3 is 47 and that by NTCIR4 is 35. We use both assignments. So, the total number of the assignments is 82. We use one community where 50 IR agents and one PA exist. We divide the rated documents evenly and randomly to each IR agent. So, each IR agent has about 3000 documents. The IR agent is also given to 20 assignments selected randomly from 82 ones. Each IR agent extracts keywords in the $CONCEPT$ field of each given assignment and uses the keywords as its query instead of the assignment itself. Thus 1000 queries will totally be issued. This means each query will averagely be issued 12 times. Each IR agent issues a query to target IR agents in order. We call the duration that all IR agents issue a query, a round. While an IR agent issues 20 queries in the experiments, the changes of Mean Average Precision (MAP) values of our proposed methods from the 1st round to the 20th round are investigated. The number of target IR agents is set to 10. Each target IR agent returns at most 10 documents to the query-sender IR agent in descending order according to their scores.

Figure 1 depicts the results comparing the MAP values of the following 5 methods: (A) Query-Multicasting with User Feedback and the QL1 (QM-wQL1), (B) Query-Multicasting with User Feedback and the QL2 (QM-wQL2), (C) Query-Multicasting with User Feedback, but employing neither the QL1 nor QL2 (QM-wUF), (D) Centralized server-type IR method with User Feedback (CM-wUF) and (E) Query-Multicasting without User Feedback (QM-woUF). In the figure, we can see the effects of user feedbacks and the two query learning methods, comparing (E) with other 4 methods.

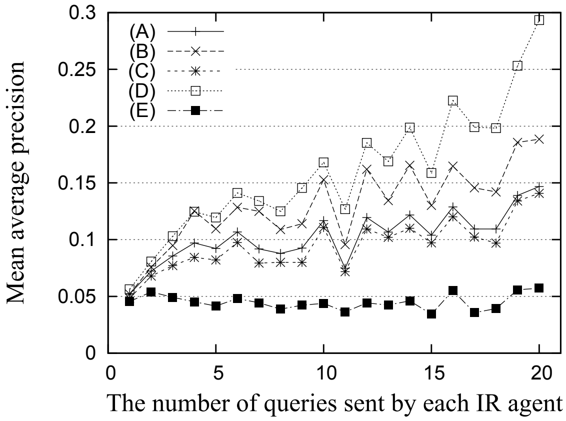

Figure 1: The Comparison of Query-Learning Methods.

The reason why the difference between QM-wUF and QM-wQL1 is small, the QM-wQL1 method only uses the latest record to a query in the Q/MRH, and selects one target IR agent in the $from$ field of the record and the others by multicast except for the agents in the $list-of-target$ field of the record. On the other hand, the QM-wQL2 repeatedly issues a query to different target IR agents selected in the $list-of-responder$ field, with making the query-sender IR agent accumulate the retrieval results on the query. The MAP value of the QM-wQL2 is almost equal to that of the CM-wUF until the 5th round, but gradually gets smaller and smaller after that. It is because the CM-wUF can accumulate negative judgement information, which are the information about irrelevant documents, and avoid reevaluation of the documents. On the other hand, the QM-wQL2 just shares positive judgement information in a community since it makes a target IR agent return to a query-sender IR agent relevant documents and unevaluated documents which may be evaluated as irrelevant ones.

4. CONCLUSIONS

This paper discussed the methods for improving the ACP2P method and showed the effects of user feedbacks and query-learning methods through experimental results. Investigating a method using negative feedback is future work.

5. ACKNOWLEDGEMENTS

We used NTCIR3 and NTCIR4 WEB Test Collections under permission from National Institute of Informatics. We deeply thank Mr. Hirotake Kobayashi for his great support to the experiments.

6. REFERENCES

[1] T. 3rd and 4th NTCIR Workshop Data. http://research.nii.ac.jp/ntcir/{ntcir-ws3/work-en.html#data,ntcir-ws4/data-en.html}.

[2] T. Mine, D. Matsuno, A. Kogo, and M. Amamiya. Design and implementation of agent community based peer-to-peer information retrieval method. In CIA2004, volume LNAI 3191, pages 31–46, 9 2004.

[3] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In NIST Special Publication 500-226: TREC-3, 1994.