A Comparative Evaluation of Two User Feedback Techniques for Requirements Trace Retrieval

Yonghee Shin and Jane Cleland-Huang
Center of Excellence for Software Traceability (CoEST)
School of Computing, DePaul University
Chicago, IL, USA
yshin@cdm.depaul.edu, jhuang@cs.depaul.edu

ABSTRACT

In automated requirements trace retrieval, significant improvements can be realized through incorporating user feedback. In this paper we introduce a relatively new technique named Direct Query Manipulation (DQM) and compare its effectiveness against Rocchio, the current defacto standard for integrating user feedback into automated tracing methods. The two techniques are evaluated empirically through a series of simulations and a user study, conducted by tracing requirements for WorldVista, an electronic healthcare information system against requirements from the Certification Commission for Healthcare Information Technology. Our results show that both Rocchio and DQM return significant improvements in trace quality in comparison to the vector space model, a fully automated technique. DQM performs slightly better than Rocchio in terms of trace quality with minimal difference in human effort. The hybrid approach provides further improvement over both individual approaches of DQM and Rocchio.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications; D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement

General Terms

Experimentation, Measurement, Documentation

Keywords

Traceability, Query modification, Rocchio, Relevance feedback

1. INTRODUCTION

Requirements traceability is an essential component of the software engineering lifecycle for most non-trivial software intensive systems. It supports numerous software engineering activities such as impact analysis, compliance verification, and coverage analysis [12], and is often mandated in a project by governmental regulations, or as part of a process improvement initiative. In many projects, the number of required traces across different types of artifacts can grow very large [11], causing the manual effort required to establish and maintain such traces to be inhibitive.

Several different researchers have attempted to address these problems by using information retrieval (IR) methods to automatically generate trace links. The most popular approaches have utilized either the Vector Space Model (VSM) [8], probabilistic approaches [5], or Latent Semantic Indexing (LSI) [3]. Although these methods can significantly reduce the tracing effort in most projects, the generated traces are usually quite imprecise and require an analyst to spend significant time evaluating the results in order to find the correct set of links [7].

One promising solution is to incorporate user feedback into the process of automated trace retrieval. Hayes et al. [8] and De Lucia et al. [9] used the standard information retrieval technique of Rocchio [4] relevance feedback to improve trace quality. The Rocchio algorithm utilizes relevance feedback captured from an initial set of trace links to modify the underlying representation of the query. It has been shown to significantly improve trace quality. Cleland-Huang et al. [6] introduced the Direct Query Modification (DQM) approach, which allows a user to directly modify the trace query by adding terms and filtering out unwanted terms.

Although several researchers have previously evaluated the use of Rocchio in the tracing process, there has previously been neither a rigorous evaluation of DQM nor a comparison between Rocchio and DQM. We therefore first conducted an experiment to compare Rocchio and DQM using simulated user feedback and then performed a user study utilizing actual user feedback. Additionally, we investigated whether combining the two approaches could improve trace quality. We evaluated the techniques against two requirements data sets: the requirements from the Certification Commission for Healthcare Information Technology [1], and the requirements for WorldVistA [2], an electronic healthcare information system.

The contributions of this paper are three-fold. First, we demonstrate that DQM performs better than Rocchio, which is currently considered the defacto standard for integrating user feedback into the trace retrieval process. Second, we demonstrate that the hybrid approach of Rocchio and DQM provides a noticeable improvement in trace quality over both

SAC '12, March 26-30, 2012, Riva del Garda, Italy.

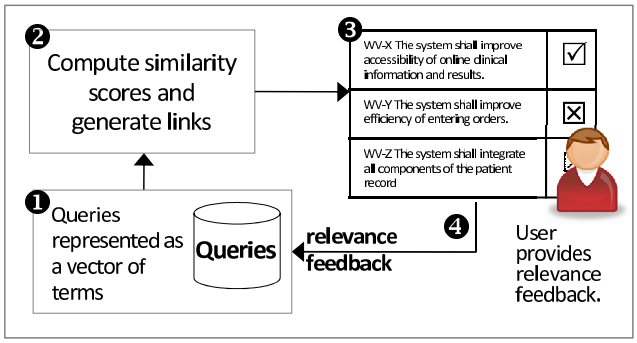


Figure 1: Rocchio process

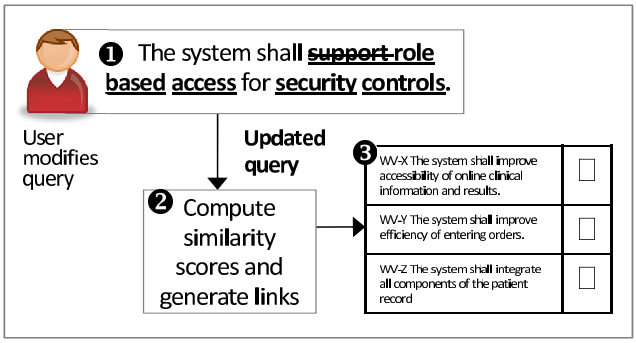


Figure 2: Direct Query Modification process

individual techniques. Third, our user study identifies requirements for feedback-based trace retrieval techniques and therefore provides directions for future research.

The remainder of the paper is organized as follows. Sections 2 and 3 explain Rocchio and DQM techniques in detail. Sections 4 and 5 report comparative analyses of Rocchio and DQM using the simulated and actual user feedback, respectively. In section 6, we propose and evaluate a hybrid approach of DQM and Rocchio. Section 7 discusses threats to validity and section 8 summarizes and discusses future work.

2. ROCCHIO FEEDBACK

To describe the Rocchio algorithm [4], we first briefly describe the Vector Space Model (VSM), which is a basic algorithm for computing the similarity between two documents. We then explain how Rocchio modifies the VSM to incorporate user feedback and to potentially improve tracing results. In the VSM, each query and each document is represented as a term vector defined in the space of all terms found in the set of queries $T = t_1, t_2, ..., t_n$. More formally, a document d is represented as a vector $\vec{d} = (w_{1,d}, w_{2,d}, ..., w_{n,d})$, where $w_{i,d}$ represents the term weight associated with term i for document d. A query is similarly represented as $\vec{q} = (w_{1,q}, w_{2,q}, ..., w_{n,q})$. Each term t is assigned a weight using a standard weighting scheme known as $tf - idf$ [4], where tf represents the term frequency, and idf the inverse document frequency. Term frequency with respect to document d is often computed as $tf(t_i, d) = (freq(t_i, d))/(|d|)$, where $freq(t_i, d)$ is the frequency of the term in the document, and $|d|$ is the length of the document. Inverse document frequency idf, is usually computed as $idf_{ti} = log_2(n/n_i)$ where n is the total number of documents in the traceable collection, and n_i is the number of documents in which term t_i occurs. The individual term weight for term i in document d is then computed as $w_{i,d} = tf(t_i, d) \times idf_{ti}$.

A similarity score $sim(d, q)$ is computed between document d and query q as the cosine of the angle between the two vectors as follows:

$$sim(d,q) = \frac{(\sum_{i=1}^{n} w_{i,d} w_{i,q})}{\left(\sqrt{\sum_{i=1}^{n} w_{i,d}} \cdot \sqrt{\sum_{i=1}^{n} w_{i,q}}\right)} \quad (1)$$

Any query-document pair receiving a similarity score over a threshold value is treated as a candidate trace link. Given a query vector q and a set of document vectors D_q returned as a result of issuing trace query q, the Rocchio algorithm assumes that an analyst has reviewed D_q and separated the results into two subsets D_{rel} consisting of all document vectors that are specified by the analyst as relevant to q, and D_{irr} consisting of all document vectors that are specified as irrelevant to q. D_{rel} and D_{irr} are disjoint and will not cover the entire set D_q, unless the analyst has provided complete relevance feedback on all possible links for query q.

Standard Rocchio techniques modify the query vector based on relevance feedback as follows:

$$\vec{q}_{mod} = (\alpha\, \vec{q}_{original}) + (\beta\, \vec{q}_{rel}) - (\gamma\, \vec{q}_{irr}) \quad (2)$$

where

$$\vec{q}_{rel} = \frac{1}{|D_{rel}|} \sum D_{j \in D_{rel}} d_j \quad (3)$$

and

$$\vec{q}_{irr} = \frac{1}{|D_{irr}|} \sum D_{k \in D_{irr}} d_k \quad (4)$$

which intuitively takes the initial query vector q, and adds the weighted vectors of relevant documents in D_{rel}, and subtracts the weighted vectors of documents in D_{irr}. Weighting factors α, β, and γ are used to assign different emphasis to the initial query vector, positive feedback, and negative feedback, respectively.

To execute Rocchio, the top n candidate links are presented to the analyst and relevance feedback is then elicited. The query vector $\vec{q}_{mod}$ is recomputed and the IR query is rerun using the new term weights in order to generate a new list of candidate links. The process is repeated iteratively until the analyst is satisfied with the resulting trace links or no further improvements are realized. The Rocchio process is illustrated through steps 1-4 in Figure 1.

3. DIRECT QUERY MODIFICATION

DQM captures an entirely different type of feedback from the analyst. Instead of asking the analyst to specify whether a set of candidate links are relevant or not, DQM allows the analyst to directly manipulate the trace query by adding terms or filtering out unwanted terms. After the analyst modifies a query and reruns the query, the documents are presented to the analyst in the order of relevance. The query modification is repeated iteratively until the analyst is satisfied. Figure 2 illustrates the DQM process. Note that the check boxes for Step 3 in Figure 2 are used only for documentation purposes and are not actually used to compute similarity score as they were in Figure 1 for the Rocchio process.

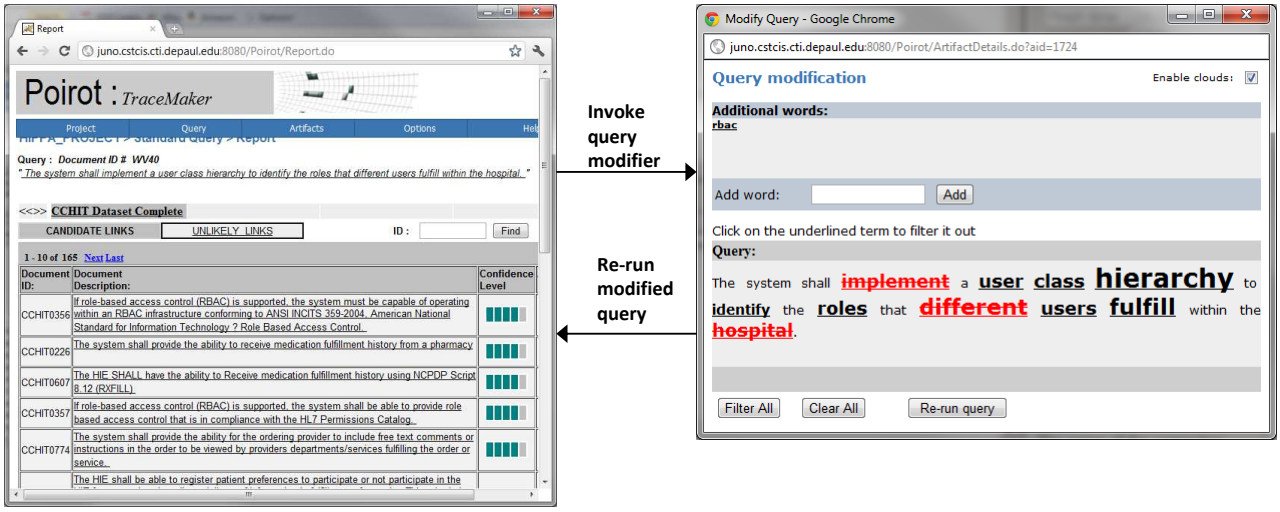


Figure 3: The Poirot tool to iteratively modify a trace query by eliminating and adding new terms

4. EXPERIMENT #1: SIMULATED USER FEEDBACK IN ROCCHIO AND DQM

An experiment was conducted to compare trace quality from the Rocchio and DQM approaches. For this experiment we followed the standard approach for evaluation of the Rocchio algorithm by simulating user feedback based on traces in a previously constructed answer set.

4.1 Datasets

The experiments were conducted using requirements for WorldVistaA, an electronic healthcare information system developed by the USA Veterans Administration against the requirements for Ambulatory, Health Information Exchange, developed by the Certification Commission for Healthcare Information Technology (CCHIT). WorldVista contains 116 requirements, of which 72 were traced to CCHIT requirements. We used those 72 requirements as a query set. The CCHIT EHR specification includes 1,064 requirements designed to evaluate healthcare information systems for certification purposes. Four researchers at DePaul University developed trace links from WorldVista to CCHIT requirements. All traces were manually evaluated by four researchers on the team, and the resulting traces were further validated and refined through a series of formal technical reviews. The resulting traceability matrix was then treated as the "answer set" for the experiments in this paper.

4.2 Evaluation Metric

A good trace retrieval technique should rank relevant documents higher than non-relevant documents to minimize the time needed by an analyst to evaluate and use the retrieved links. *Average precision* [10] measures how well the relevant documents are ranked at the top of the retrieved links and is computed as follows:

$$AveragePrecision = \frac{\sum_{r=1}^{N}(P(r) * relevant(r))}{|RelevantDocuments|} \quad (5)$$

where r is the rank of the requirement in the ordered set of candidate trace links, N is the number of retrieved documents, $relevant()$ is a binary function assigned 1 if the rank is relevant and 0 otherwise, and $P(r)$ is the precision computed after truncating the list immediately below that ranked position. Average precision returns a higher value (approaching 1) when more relevant documents are retrieved towards the top of the ranked list according to the computed similarity score. Because we compute average precision for all true links, our use of average precision implies a recall of 100%. As a single metric to evaluate a trace retrieval technique, we use *Mean Average Precision (MAP)* which averages the average precision from all queries.

4.3 Experimental Design

Traces were generated from the WorldVista requirements to the CCHIT certification requirements using each of the following techniques.

4.3.1 Baseline

The original WorldVista requirements were used as the trace queries, and candidate trace links were generated against CCHIT certification requirements using the vector space model described in section 2. MAP was then computed by comparing the generated links against the answer set.

4.3.2 Rocchio

To evaluate Rocchio, candidate trace links were first generated as in the baseline experiment. For each WorldVista requirement, we simulated presenting the ten highest ranked candidate links to the user. Following standard experimental practices, if a link was documented in the answer set, the feedback was recorded as 'relevant', and if it was not documented in the answer set, it was recorded as 'non-relevant'. In each iteration we simulated the task of presenting the ten top ranking candidate links, for which no feedback had been previously elicited, to the user for relevance feedback. Following each iteration, term vectors for each query were modified using equation (2) with α=0.8 and β=0.2 and γ=0. Based on preliminary experiments we found Rocchio works best when gamma is set to zero, i.e. when only positive feedback is considered. Each subsequent iteration used the updated query vector to regenerate trace links, and MAP

was computed at the end of each iteration. The experiment continued until the improvement in MAP from previous iterations was less than 0.01.

4.3.3 DQM

To evaluate DQM, a team of three researchers at DePaul university manually traced each of the WorldVista requirements to CCHIT using our automated tracing tool, Poirot. The Poirot tool allows users to modify queries by removing or adding terms. This is illustrated in Figure 3, which shows that a trace user has eliminated the terms of *implement*, *different*, and *hospital*, and added the term of *RBAC*. Initially, the Poirot tool presents documents ranked in descending order of similarity score. When the user re-executes the trace using the modified query, Poirot returns the whole set of documents again in the order of newly computed similarity score. By browsing the returned documents, the user can immediately check whether the query modification was effective in improving trace quality.

The researchers simulated the way in which actual users interact with Poirot, by trying different combinations of filtered and additional terms until they were satisfied with the tracing results. Trace links were then generated for the final version of modified queries using the vector space model described in section 2.

4.4 Results

Figure 4 shows the distribution of the average precision from these experiments using boxplots. The basic approach, which used the unmodified WorldVista requirements as the trace queries, resulted in MAP of 0.36. The Rocchio process continued through five iterations (R1 through R5 in Figure 4) reaching MAP of 0.48 . In general, the greatest improvements were seen in the first three iterations, with slightly less prominent improvements in subsequent iterations. Finally, the DQM approach returned MAP of 0.53, which was 5% higher than the Rocchio results.

A correlation analysis was performed to analyze the relationship between the number of correct trace links for a query and the resulting average precision value, and the Spearman correlation coefficient was computed at 0.43 for Rocchio and only 0.22 for DQM. This suggests that Rocchio performs best in cases where a single trace query has many correct links. This makes sense given that positive feedback has been shown to be most effective when Rocchio is implemented in the trace retrieval domain. At the same time, the majority of trace queries have only a few correct links. For example, in our data set, only 28% of queries had more than ten trace links (See Table 1). This explains why DQM is generally more effective than Rocchio.

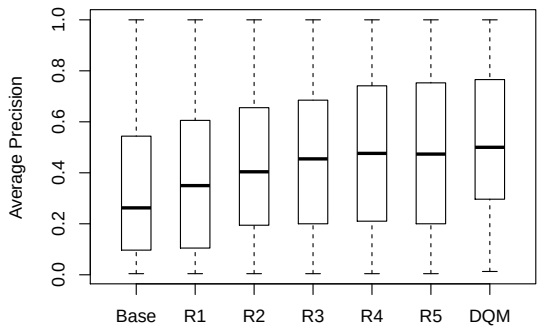


Figure 4: Results from simulated study

Table 1: Distribution of trace links

Number of trace links	Number of queries (%)
<= 10	52 (72%)
10 < N <=20	11 (15%)
20 < N <= 30	9 (13%)

5. EXPERIMENT #2: ROCCHIO VERSUS DQM USING ACTUAL USER FEEDBACK

The overall goal of a trace retrieval technique is to deliver high quality traces with minimal human effort. The previous experiment demonstrated that DQM provided slightly better quality results than Rocchio. In this experiment, we investigated whether the two techniques require similar human effort. For this purpose, we performed a user study in which the previous simulation was replaced by actual human feedback. The Rocchio study was conducted using TraceLab, which is a tool for modeling and executing trace retrieval experiments, and was developed under an NSF Major Research Instrumentation grant by researchers at DePaul University and other collaborating universities. We configured the Rocchio experiment in TraceLab, including the GUI component depicted in Figure 5 which was designed to elicit Rocchio-style feedback. This component displays ten retrieved documents in each iteration and allows each user to determine how many iterations of feedback to provide for a given trace query. DQM was evaluated using our Poirot tracing tool, which also displays ten retrieved documents per page.

5.1 Experimental Design

Eight members of the Software and Requirements Engineering Lab at DePaul University participated in this experiment. Each participant traced WorldVista requirements to CCHIT requirements, and completed five traces using Rocchio, and five using DQM. Because the participants were not familiar with the requirements for electronic healthcare information systems, the speed of the tracing process was expected to improve as the participants completed some initial queries. To reduce this learning effect, we allowed the participants to read all the queries before the experiments began in order to increase their comprehension of the domain. The order of usage of techniques and data sets can also affect the results. Therefore, we assigned two participants to two different orders of technique usage and two data sets. Table 2 shows the assignment of the participants to each combination of experimental setting.

Table 2: Design for user study

User	First study	Second study	Query set for first study	Query set for second study
User1, User5	DQM	Rocchio	1	2
User2, User6	Rocchio	DQM	1	2
User3, User7	DQM	Rocchio	2	1
User4, User8	Rocchio	DQM	2	1

5.2 Results

The results showed that the minimum number of Rocchio iterations per query was two, and the maximum number was nine. Figure 6 shows the distribution of the time taken for

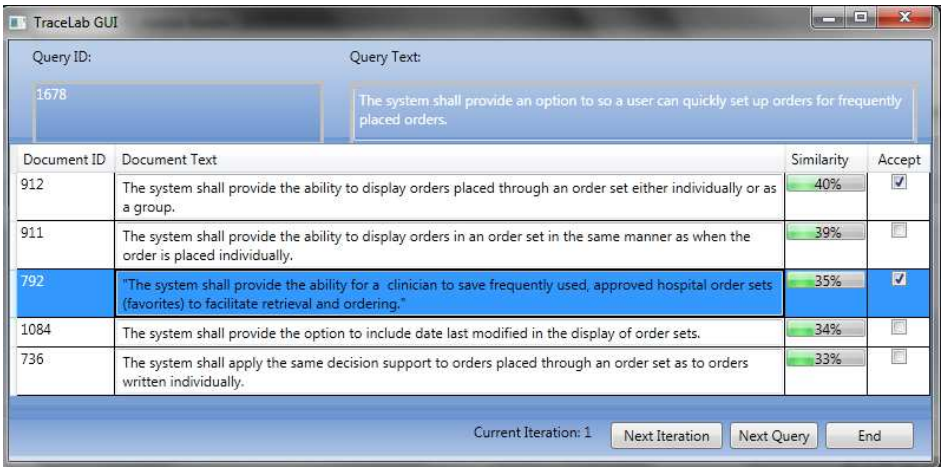
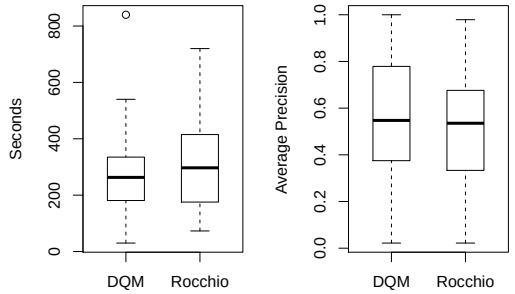


Figure 5: Rocchio component in TraceLab

the experiment and the average precision. For DQM, tracing a query took an average of 5 minutes and 17 seconds with a minimum of 30 seconds and a maximum of 14 minutes across all queries for all participants. For Rocchio, tracing a query took an average of 5 minutes and 36 seconds with a minimum of 1 minute and 30 seconds and a maximum of 16 minutes. Tracing with DQM took an average of 20 seconds shorter than with Rocchio, but this difference was not statistically significant based on a t-test significance level of 0.05.

In terms of trace quality, the MAP for DQM for the eight participants was 0.57 with a minimum average precision of 0.31 and a maximum of 0.77. The MAP for Rocchio was 0.54 with a minimum average precision of 0.46 and a maximum of 0.66. Therefore, DQM provided 3% higher MAP, but again the difference was not statistically significant at a significance level of 0.05 with the t-test.



(a) Time (b) Average Precision

Figure 6: Results from user study

A qualitative analysis was also performed in which each of the participants was asked which technique they preferred and the reason for their preference. These questions were asked after the user had completed the study. Four users preferred DQM to Rocchio. Among them, three users liked the fact that DQM gave them more control over the selection of important terms, and two users liked the fact that DQM quickly returned updated retrieval results based on the modified queries. This gave the users useful feedback on whether their changes to the queries were effective in improving the tracing results or not. On the other hand, four users preferred the simplicity of Rocchio, claiming that it enabled them to focus on the more basic task of determining whether the small set of presented links were correct or not, and did not require them to figure out how to improve the queries themselves. However, the users who preferred DQM over Rocchio, complained that Rocchio did not provide any quantitative measures to inform them whether their feedback had improved the tracing results or not. An interesting observation is that in general the participants who preferred Rocchio spent longer to complete the ten trace queries than those who preferred DQM. The Spearman correlation coefficient between the two feedback types and the total tracing time was 0.79 at 0.05 significance level. Slow readers seem to prefer the Rocchio approach because each document is presented to them only once. More interestingly, six out of the eight participants provided better feedback for the techniques they preferred resulting in higher MAP, indicating feedback methods should consider the reading style of the users.

Overall, this user study reveals important requirements for incorporating user feedback into the trace retrieval process. First, several users wanted to have more control over the terms they deemed important for tracing. Second, users wanted more visual feedback to keep them informed of how their tracing effort was progressing; and finally some users performed better when less cognitive effort was required. Confirming these requirements for industrial users in order to improve the feedback methods and tools will be included in our future work.

6. EXPERIMENT #3: HYBRID APPROACH

Although Rocchio and DQM produced similar improvements, they accomplished this in slightly different ways. Whereas Rocchio gradually increases or decreases the weighting of terms, DQM allows the user to add new terms or delete terms entirely from the query. This means that even though Rocchio improves the trace quality over the baseline model, the queries still can contain unimportant terms, degrading the quality of generated traces. Therefore, we investigated whether combining the two approaches in a synergistic manner could improve trace quality. Although there are several different ways for combining DQM and Rocchio, we adopted a hybrid model in which the analyst first modifies a query using the DQM process, and then performs the Rocchio pro-

cess by providing relevance feedback. An experiment was conducted to compare the hybrid approach and the individual approaches of Rocchio and DQM. In this experiment, the user feedback was simulated using the technique described for Experiment #1.

Figure 7 presents the distribution of average precision from the hybrid approach together with the results from Experiment #1. The hybrid approach returned MAP of 0.58 which is 5% higher than the MAP from DQM and 10% higher than that from Rocchio. This result shows that the hybrid approach can effectively improve accuracy of tracing results. We leave a user study of this approach to future work.

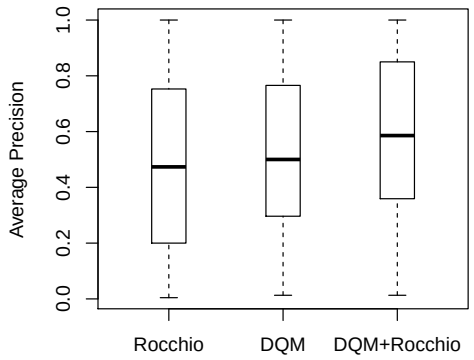


Figure 7: Results from hybrid approach

7. THREATS TO VALIDITY

Different metrics can measure different aspects of trace retrieval techniques. In our study, we were interested in the ability to rank relevant documents at the top of the the retrieved links. To deal with this construct validity, we used mean average precision which is a well-accepted measure for this purpose. As it is extremely effort intensive and laborious to establish test sets for such a study, we were only able to conduct experiments on a single dataset. Therefore, to provide external validity for generalizing our observations, further replicated studies with additional data sets are required. We are currently collecting additional data sets. Clearly using only eight participants and five queries for each technique in the user study is not enough to provide a significant statistical evidence and to provide conclusion validity, and we intend to replicate the study with a larger group of participants. On the other hand, both the simulated study and the user study provide consistent results supporting the reliability of our conclusion. Despite these limitations, we believe our observations in this study contribute to the body of knowledge in traceability research by providing empirical evaluation results of two feedback approaches, by demonstrating the superiority of the hybrid approach, and by guiding future research directions.

8. CONCLUSIONS

This paper has described a study of two different user feedback methods and has returned some interesting results. First, it has shown that DQM performs slightly better than Rocchio in terms of accuracy and requires very similar effort. This is interesting given the fact that Rocchio is broadly accepted as the standard approach for incorporating user feedback into the trace retrieval process, and suggests that tools such as Poirot, which integrate DQM, can be as effective as tools utilizing Rocchio. Furthermore, our study has identified several factors that impact the effectiveness of each technique. For example, Rocchio unsurprisingly performs best in trace queries with multiple correct links, while DQM performs equally well regardless of the number of correct links for a query. Our experiments also showed that users who completed the tracing tasks relatively quickly, preferred DQM over Rocchio, and vice versa, and this observation certainly warrants additional investigation. Finally, our results demonstrate the potential for using a hybrid approach to quite significantly improve accuracy of the tracing results. Future work will involve replication of these experiments to evaluate the two techniques across different kinds of tracing domains with industrial users.

9. ACKNOWLEDGMENTS

The work described in this paper was primarily funded by U.S. National Science Foundation grant #CCF-0810924. The experiments were conducted using TraceLab, developed under grant #CNS-0959924.

10. REFERENCES

[1] Certification commission for health information technology, http://www.cchit.org.

[2] Vista Electronic Health Record and Health Information System, http://www.worldvista.org.

[3] G. Antoniol, G. Canfora, G. Casazza, A. D. Lucia, and E. Merlo. Recovering traceability links between code and documentation. *IEEE Trans. Software Eng.*, 28(10):970–983, 2002.

[4] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[5] J. Cleland-Huang, B. Berenbach, S. Clark, R. Settimi, and E. Romanova. Best practices for automated traceability. *IEEE Computer*, 40(6):27–35, 2007.

[6] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker. A machine learning approach for tracing regulatory codes to product specific requirements. In *ICSE (1)*, pages 155–164, 2010.

[7] D. Cuddeback, A. Dekhtyar, and J. Hayes. Automated requirements traceability: The study of human analysts. *Requirements Engineering, IEEE Intn'l Conference on*, 0:231–240, 2010.

[8] J. H. Hayes, A. Dekhtyar, and S. K. Sundaram. Advancing candidate link generation for requirements tracing: The study of methods. *IEEE Trans. Software Eng.*, 32(1):4–19, 2006.

[9] A. D. Lucia, R. Oliveto, and P. Sgueglia. Incremental approach and user feedbacks: a silver bullet for traceability recovery. *Software Maintenance, IEEE International Conference on*, 0:299–309, 2006.

[10] C. D. Manning, P. Raghavan, and H. Schuätze. *Introduction to Information Retrieval*. Cambridge University Press, NY, USA, 2008.

[11] K. Pohl, R. Dömges, and M. Jarke. Towards method-driven trace capture. In *CAiSE*, pages 103–116, 1997.

[12] B. Ramesh and M. Jarke. Toward reference models of requirements traceability. *IEEE Trans. Software Eng.*, 27(1):58–93, 2001.