Abstract

Dynamic trace retrieval enables developers to generate traces between requirements and other software engineering artifacts on an as-needed basis and provides support for critical software engineering activities such as requirements validation and impact analysis. It eliminates the need to construct and maintain explicit traceability links and alleviates the extensive effort that is required to perform a manual trace when no supporting tool or trace matrix is available. Despite these advantages, trace retrieval approaches suffer from precision problems that require an analyst to evaluate the retrieved links and decide which ones are correct and which are incorrect. In this paper we describe the use of confidence scores to help the user make decisions about the correctness of each retrieved link. Initial confidence scores are determined through analyzing the distribution of true-links and non-links within three data sets and then refined through additional knowledge of actual link distribution as it becomes available for the current project. The paper also describes a small usability study that we conducted to evaluate the effectiveness of confidence scores to support the decision making process.

1. Introduction

Requirements traceability has been widely recognized as an important support mechanism for effectively managing the development and evolution of a software system. However numerous studies have shown the difficulties of maintaining traceability links explicitly using techniques such as a traceability matrix or hyperlinks [3,7,9]. For this reason many researchers have investigated the use of information retrieval methods to solve the trace retrieval problem [2,4,5,8,12]. These methods are applicable because most software artifacts such as requirements, design documents, source code, and test cases are either entirely textual in nature or contain significant amounts of textual information. Information retrieval methods allow traces to be dynamically generated rather than manually created and maintained, and therefore this approach offers an enticing solution to the problems surrounding long term maintenance of traceability links.

Unlike a Google search that typically favors precision over recall, a traceability query must favor recall. This is because if critical links are not successfully retrieved an impact analysis may be incomplete or requirements may appear to be only partially implemented in the design and code. In our work on dynamic requirements trace retrieval we therefore established a goal to achieve 90% or higher recall on all of the traces. Experimental results have shown that when recall of a dynamic trace tool reaches 90%, precision drops below 40%, sometimes even less than 10% [2,4,5,8,12]. Our previous work in enhancing precision through incorporating additional factors into the retrieval process improved precision by up to 10% in some data sets, but still resulted in precision ranging from 20% to 40% at recall values of 90% [5].

A retrieval result with such a low precision forces the user to evaluate the set of retrieved links (known as candidate links) in order to determine which traces are correct and which ones are incorrect. This introduces the opportunity for human error and requires effort on the part of the human analyst. However, given the fact that many if not most organizations have almost negligible traceability practices in place, the alternative is to either rely on expert judgment to retrieve impacted artifacts or to perform a time-consuming, error-prone manual evaluation of the entire document collection.

This paper describes a support mechanism to help the analyst decide which candidate links are relevant to the requirement being traced. A confidence score is displayed on the results screen representing the confidence a user could have that each link is in fact a true link. Although the use of confidence scores within the information retrieval domain is not new, the importance of implementing this concept to support requirements trace retrieval is critical in light of the precision problem affecting trace retrieval.

As an initial evaluation of the effectiveness of this confidence score mechanism, a small informal usability study was conducted using our Poirot:Tracemaker tool
to compare the analyst’s performance in conducting trace tasks when supported by confidence scores versus tasks that were unsupported.

The paper is organized as follows. Section 2 introduces the primary ideas behind the use of confidence scores and then describes a method for computing confidence scores for both retrieved and rejected links. Section 3 describes the usability study we conducted and section 4 discusses conclusions and areas of future work.

2. Confidence Scores

In a typical dynamic requirements trace the retrieval algorithm returns a similarity score representing the likelihood of a link existing between the requirement and each potentially traced artifact. A threshold value is determined and all pairs of artifacts above this value are presented to the user as candidate links for evaluation.

Unfortunately similarity scores do not tend to provide intuitive support to the user. This is illustrated by the scores returned by our probabilistic tool [4,5]. These scores ranged between 0 and 0.05 with actual scores such as 0.01027493 and 0.01134955. Optimal cutoff points were dependent upon the type of artifact being traced [11], and tended to be in the range of 0.001 to 0.02. An analyst looking at such a low score would be unlikely to intuitively believe that the link represented a true trace.

In this study confidence scores were created to provide a meaningful reference to the user when presenting the retrieval results. The scores represent a measure of belief in the correctness of the results returned by the automatic trace retrieval algorithm. They are calculated according to the probability value of a link in respect to the general distribution of probability values for true-links and false-links in the dataset. For the purposes of this study, the link distribution was initially learned from the distribution of links in three different datasets. These data sets include the Ice-Breaker system (IBS), Event Based Traceability (EBT), and Light Control System (LC) all of which are more fully described in [5].

For each pair \((d,q)\), the confidence value suggests how confident we are that a trace between a document \(d\) and query \(q\) should be established, or in other words that the link between \(d\) and \(q\) is a true link. Confidence scores are set for convenience between 0 and 100%.

In prior empirical studies that were conducted against the three data sets (IBS, EBT, and LC), similar patterns were observed among the distributions of the probability scores for true links. The results consistently showed that candidate links with larger probability scores are more likely to be true links. On the other hand, the links with very small probability scores are very likely to be non-links. This is clearly depicted in Figure 1, which shows the probability score distribution for true links against the one for non-links for the IBS system [4].

In the region in the center, true-links and non-links are mostly cross-distributed, making it difficult to discriminate between these types of links based only on the probability scores.

Figure 2 shows the changes in true links frequency in relation to the probability scores distribution. The graph shows that the frequency of true links increases with the probability scores in a non-linear fashion. In the middle region, the frequency of true links increases at a slower rate than in the top region where the probability values are higher. Similarly for links with very low probability values – i.e. within the region where the true links are scarcer - the frequency of true links increases even more slowly.

Moreover, since the threshold is typically set at a low value, we can assume that many of the links just above the threshold are likely to be non-links and therefore should be assigned low confidence scores. Similarly, the rejected links that are below the low threshold value will have even lower confidence scores. Thus users can be almost sure that these links are actually non-true links.

![Figure 1. Areas of high and low confidence](image1.png)

![Figure 2: True links density of IBS dataset](image2.png)
Based on these observations, we defined a segmentation method for computing confidence scores by considering the frequency distribution of true links in different probability intervals.

### 2.1 Segmented Confidence Scores

The segmentation method divides the range of probability scores into K+1 intervals and defines a piecewise function that computes confidence scores in each interval according to the frequency of true links in that interval.

Figure 2 illustrates the frequencies of true links for the IBS dataset in equally spaced intervals for the probability scores defined in terms of percentile ranks. The graph shows that true links frequency is relatively high when probability scores are above the 80th percentile. While for scores below the 20th percentile, the frequency of true links is quite low. The distributions of true links for EBT and LC datasets reflect a similar pattern.

The frequency of true links in the i-th segment \((p_{i-1}, p_i)\) is defined as:

\[
\mu_i = \frac{\text{number of true links in } (p_{i-1}, p_i)}{\text{number of candidate links in } (p_{i-1}, p_i)}
\]

The measure \(\mu_i\) can be regarded as the local precision metric relative to the i-th segment.

Let \(C(p)\) be the confidence score associated to a candidate link with probability score \(p\).

The 100\% confidence score is assigned to the candidate link that has probability value equal to or higher than 0.5, i.e. \(C(p)=1\) for \(p \geq 0.5\). This is based on our observation that any link with a probability value as high as 0.5 is extremely likely to be a true link. Also we assign zero confidence to pairs \((d,q)\) with zero probability score, i.e. \(C(p)=0\) for \(p=0\).

Let \(p_{\text{max}}\) be the maximum probability score returned by the tool. The set of probability scores between \(p_0=0\) and \(p_{\text{max}}\) is divided into \(K+1\) intervals. Each interval \((p_{i-1}, p_i)\), for \(i=1,\ldots,K+1\), is defined by selecting \(K\) percentile values \((p_1,p_2,\ldots,p_K)\) from the distribution of the probability scores. The \(K\) percentile values can be determined according to some prior knowledge of the true links distribution. Such prior knowledge could be based for instance on a set of known traces that can be generated without the need for user evaluation.

If no prior knowledge is available on a certain system, a simpler version of the segmentation algorithm is proposed. However, as users apply the retrieval tool to that system, information on the correctness of some links is obtained through user’s feedback, and can be used to learn the confidence values more accurately.

Once the \(K\) percentile values have been selected, the average probability \(\bar{p}_i\) of the probability scores in \((p_{i-1}, p_i)\) is computed for each interval \((p_{i-1}, p_i)\) for \(i=1,\ldots,K+1\).

The graph in Figure 3 shows an example of the piecewise function for a simple case with \(K=2\). In each segment, the confidence value \(C(p)\) is computed as a linear function of \(p\) with slope depending on the increase in true link frequency between two consecutive intervals.

The general expression of the piecewise function for computing confidence values is displayed in Table 1.

### 2.2 Confidence Scores with no prior knowledge

If no prior knowledge on the true link distribution is available, the piecewise function can be defined using a coarser segmentation for the range of the probability scores by setting \(K\) equal to two. The selected percentiles for the probability scores are just the 20\% percentile and the 80\% percentile.

Table 2 displays the average true links density for the three segments and the corresponding confidence values for the average probability scores. Such values are derived from our experimental studies.

At first, if the retrieval tool is applied to a new system, confidence values can be specified using the coarser segmentation method that uses the values in table

![Figure 3. Confidence value vs. Probability in IBS](image)

The confidence score \(C(\bar{p}_i)\) is assumed to be proportional to the true link frequency in the \(i\)-th interval, and is computed as \(C(\bar{p}_i) = \lambda \mu_i\), where \(\lambda\) is a rescaling factor that can be set by the user.

Each segment in the piecewise function is defined as a straight line joining pairs of points defined as \((\bar{p}_i, C(\bar{p}_i))\) and \((\bar{p}_{i+1}, C(\bar{p}_{i+1}))\) for \(i=0,\ldots,K\). The graph in Figure 3 shows an example of the piecewise function for a simple case with \(K=2\). In each segment, the confidence value \(C(p)\) is computed as a linear function of \(p\) with slope depending on the increase in true link frequency between two consecutive intervals.

The general expression of the piecewise function for computing confidence values is displayed in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Linear stepwise function to compute confidence values</th>
</tr>
</thead>
</table>
| \[
| C(p) = \begin{cases} 
| \frac{C(\bar{p}_i)}{\bar{p}_i} \times p & \text{for } p \in (0, \bar{p}_i) \\
| \frac{C(\bar{p}_i) + C(\bar{p}_{i+1}) - C(p)}{\bar{p}_{i+1} - \bar{p}_i} \times (p - \bar{p}_i) & \text{for } p \in (\bar{p}_i, \bar{p}_{i+1}) \\
| 1 - \frac{1 - C(\bar{p}_{i+1})}{0.5 - \bar{p}_i} (0.5 - p) & \text{for } p \leq \bar{p}_i \\
| \end{cases} \quad \text{for } 2 \leq i \leq K+1 
| \]
| \[
| C(\bar{p}_i) = \frac{C(\bar{p}_i) + C(\bar{p}_{i+1}) - C(p)}{\bar{p}_{i+1} - \bar{p}_i} \times (p - \bar{p}_i) & \text{for } p \in (\bar{p}_i, \bar{p}_{i+1}) \\
| 1 - \frac{1 - C(\bar{p}_{i+1})}{0.5 - \bar{p}_i} (0.5 - p) & \text{for } p \leq \bar{p}_i \\
| \end{cases} \quad \text{for } 2 \leq i \leq K+1 
| \]
2. As user’s feedback becomes available, a more refined segmentation method can be specified using the equations in Table 1. The user’s feedback can be used to build a set of known traces that can be used to estimate the distribution of the true links with respect to the probability scores.

3. Assessing support for trace evaluation

We conducted a small usability study to help us understand how analysts use confidence scores when evaluating trace retrievals and to serve as a pilot study for assessing the added value of confidence scores.

The six participants were all graduate students at DePaul University. All six participants had completed a course in Requirements Engineering (SE482) and/or one in Software Engineering (SE425) and so had acquired knowledge of the software development lifecycle and typical industrial practices, as well as a good understanding of the purpose and mechanisms of requirements traceability. There were three males and three females. Five of the participants had real work experience in the IT industry ranging from six months to five years, with an average of 2.2 years. Work positions included programmer, tester, and project manager. None of the participants had been responsible for performing impact analysis tasks in their organizations.

3.1 Procedure

During the study the participants were asked to evaluate the links returned by four queries. As depicted in Figure 4, the description of the query is provided at the top of the screen, and the retrieved documents (in this case class diagrams) are listed below. The analyst may click on the class ID in order to view the actual content of the class (ie method names and attributes). A horizontal bar demarcates the previously defined threshold established to achieve recall goals of 90%. The default value of the “Accept” box is checked for links above the threshold and unchecked for links below the threshold.

The study used a repeated measure design where each participant received two treatments. Treatment one, served as a control, and consisted of two retrieval result pages without the display of confidence scores. Treatment two consisted of two result pages that included the confidence scores. In both cases the candidate links were displayed in descending order according to the confidence levels (even though these were not always visible). The four queries were selected to represent typical recall and precision results. For each query the participant was asked to evaluate whether the requirements traces were relevant or not by making sure that the accept box was checked for relevant classes and unchecked for non-relevant ones. In many cases the default values were correctly set and the user would then just leave the check box in its current state.

Prior to each individual session, the investigator gave the participant a five-minute introduction, describing the general process and providing a brief description of the Ice breaker data set which was used for all four queries. The participant was then given a written script with the detailed explanation of the tool functionality, the concept of the traceability system, and background information describing the IBS system. Confidence levels were not mentioned during the training session. The participants were asked to speak out loud as they evaluated the queries as this could provide useful information concerning their perception of the tool and confidence scores [1].

Three participants received retrieval results without confidence scores first followed by results with confidence scores, and the other three participants were assigned treatments in the opposite order. All screen displays, user input and voice were recorded during the evaluation, and the participant was asked to complete each query within ten minutes. When the participant

<table>
<thead>
<tr>
<th>i</th>
<th>Segments</th>
<th>$\mu_i$</th>
<th>$C(\bar{P}_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0 , 20th percentile)</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>2</td>
<td>(20th percentile, 80th percentile)</td>
<td>0.15</td>
<td>0.23</td>
</tr>
<tr>
<td>3</td>
<td>(80th percentile, $p_{max}$)</td>
<td>0.4</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 2. True links density of three segments ($\lambda=1.5$)

![Figure 4. Results screen with confidence scores.](image)
completed a task in a shorter amount of time, the completion time was recorded.

After completing the evaluation, the participant was asked a few questions about his or her thoughts on the confidence scores and their responses were recorded.

### 3.2 Results Evaluation

Two measures were used to evaluate the effect of the confidence scores on the evaluation behavior of the individual participant. These were completion time and the correctness of the evaluation for each query. Links in each result page were divided into two groups. One group consisted of links for which the user accepted the default ‘accept’ value, and the other group consisted of links for which the participant changed the default value. We observed the number of links in each group and determined how many of these were correctly changed or correctly remained unchanged. These results are displayed in Table 3.

It should be noted that in the first two sessions, participants were presented only thirteen links in the result page. However one of the questions in the follow-up interview asked “What is the lowest confidence score that you consider would be worth your while to evaluate?” and one of the first participants responded that she would “evaluate every link regardless of the confidence score.” We determined that the relatively small results set may have allowed the participants to feel as if it were feasible to examine every single link, ignoring the relevance information provided by the tool. We therefore decided to reset the tool so that it returned all seventy four class diagrams in the IBS system. This simulated a more realistic workload and eliminated the effect of the reduced task on the behavior of the participants.

### 5.4 Qualitative Feedback

Following completion of the four queries the participants were each asked a series of questions. Interestingly despite the fact that the confidence scores appeared to improve the correctness of the results (see Table 3), most of the participants generally indicated that the scores had not directly helped them. In response to the question “Do you think the confidence scores were helpful in your evaluation of the links?” one of the participants responded “Actually no! As a feature it’s very helpful, but as a function it didn’t function well enough to help me. But I didn’t benefit from it at all,” while another responded that “I did not (use them), um I think at least not actively. I was mainly looking at the name, the description of the name as well as the diagram itself. It does affect you a little bit, because when you get down here (pointing to the low confidence scores at the bottom of the screen) there’s a bunch of less than 1% which psychologically could apply to an idiot (laugh) if you get what it’s saying, but consciously no, but I’m sure at some level maybe.” However one participant said that he “sort of spent less time on the <1.” Regardless of these comments, the speak-out loud protocol indicated that participants were much more prone to glance over these low confidence links. For example one user repeatedly said the word “skipping” as she glanced over them.

Our results suggest that even though the participants said that the confidence scores were not helpful, their results were generally more accurate when the confidence scores were displayed. Clearly a larger scale study is needed to determine if this hypothesis holds true over a larger study. Interestingly user # 4 did not even notice the confidence scores on the screen and therefore did not use them at all.

One of the primary problems appeared to stem from the fact that the participants wanted the confidence scores to be stronger. One participant mentioned that “I noticed that the highest one (pointing at a specific query result) was only 55%. I don't consider 55% to be overwhelmingly high so I just ignored them.” In fact the confidence scores represent the probability that a given link is a true link however several users mentioned that the reason the confidence scores didn’t help them was because they were too low to enable them to assume a link was a true link. Although in some queries the highest

<table>
<thead>
<tr>
<th>User</th>
<th>Query</th>
<th>Time to complete (Mins)</th>
<th>Correct accepted default</th>
<th>Correct change default</th>
<th>Overall correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1 (13 classes)</td>
<td>Q1</td>
<td>10</td>
<td>38%</td>
<td>20%</td>
<td>31%</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>6</td>
<td>100%</td>
<td>77%</td>
<td>85%</td>
</tr>
<tr>
<td></td>
<td>Q3 *</td>
<td>4</td>
<td>100%</td>
<td>83%</td>
<td>92%</td>
</tr>
<tr>
<td></td>
<td>Q4 *</td>
<td>4</td>
<td>67%</td>
<td>100%</td>
<td>85%</td>
</tr>
<tr>
<td>User 2 (13 classes)</td>
<td>Q1 *</td>
<td>4</td>
<td>73%</td>
<td>100%</td>
<td>77%</td>
</tr>
<tr>
<td></td>
<td>Q2 *</td>
<td>3</td>
<td>29%</td>
<td>67%</td>
<td>46%</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>3</td>
<td>40%</td>
<td>50%</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>4</td>
<td>43%</td>
<td>33%</td>
<td>31%</td>
</tr>
<tr>
<td>User 3</td>
<td>Q1</td>
<td>7</td>
<td>88%</td>
<td>50%</td>
<td>85%</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>9</td>
<td>74%</td>
<td>67%</td>
<td>73%</td>
</tr>
<tr>
<td></td>
<td>Q3 *</td>
<td>6</td>
<td>100%</td>
<td>71%</td>
<td>93%</td>
</tr>
<tr>
<td></td>
<td>Q4 *</td>
<td>10</td>
<td>88%</td>
<td>63%</td>
<td>81%</td>
</tr>
<tr>
<td>User 4</td>
<td>Q1 *</td>
<td>7</td>
<td>81%</td>
<td>100%</td>
<td>84%</td>
</tr>
<tr>
<td></td>
<td>Q2 *</td>
<td>8</td>
<td>75%</td>
<td>50%</td>
<td>73%</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>5</td>
<td>90%</td>
<td>50%</td>
<td>89%</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>7</td>
<td>89%</td>
<td>94%</td>
<td>91%</td>
</tr>
<tr>
<td>User 5</td>
<td>Q1</td>
<td>10</td>
<td>87%</td>
<td>27%</td>
<td>78%</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>10</td>
<td>86%</td>
<td>94%</td>
<td>88%</td>
</tr>
<tr>
<td></td>
<td>Q3 *</td>
<td>8</td>
<td>98%</td>
<td>76%</td>
<td>91%</td>
</tr>
<tr>
<td></td>
<td>Q4 *</td>
<td>9</td>
<td>91%</td>
<td>94%</td>
<td>92%</td>
</tr>
<tr>
<td>User 6</td>
<td>Q1</td>
<td>7</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>Q2 *</td>
<td>7</td>
<td>84%</td>
<td>83%</td>
<td>84%</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>6</td>
<td>91%</td>
<td>83%</td>
<td>90%</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>3</td>
<td>98%</td>
<td>92%</td>
<td>96%</td>
</tr>
</tbody>
</table>

* Queries marked with an asterisk included confidence scores.
ranked document has a confidence score of over 90%, there are other queries such as the one in question for which the scores are lower. In a similar case another participant mentioned that “I was surprised to see a class that was not relevant had a higher confidence score than one that was more relevant but had a lower confidence score.” Another user said “it's not like Google,” indicating a problem in believability of the results and suggesting the need for additional training to explain the expected behavior of a trace retrieval system.

The threshold bar appeared useful to the participants. For example when asked if he found the bar helpful, one participant responded that “Yes. I like the bar. It created a distinction between the ones on top and the ones on the bottom.” Another user said “when I see the bar it’s like a vision … I think that it’s (links below the bar) low priority.” These comments in conjunction with our observations that users spent very little time evaluating links below the bar indicated that it was a useful feature in the decision making process.

6. Conclusions

Although very small, the usability study provided us with useful feedback for improving tool support for the user as they evaluate retrieved links during requirements validation, impact analysis, or other similar tasks. Through observing participants’ behavior and as a result of the interview process we discovered that the confidence scores provided more help in indicating where the user should NOT spend time, than in helping the user actually evaluate the correctness of higher confidence links. The ideal behavior of an analyst is to actively evaluate all links above the threshold and then to quickly skim those below the threshold. This is in fact exactly what did happen especially in those screens supported by confidence scores. Interestingly one user even mentioned that he had learned to skip the lower links as a result of seeing the confidence scores that were available in earlier screens.

Our results suggest that the confidence scores may not be helpful in their present form but could be more usefully presented as guidelines such as “Very likely”, “Somewhat likely”, and ”Unlikely”. The threshold bar also created a useful boundary between likely and unlikely links and was therefore a useful support mechanism. Although the study reported in this paper is a preliminary one it provides useful insight into the way candidate links should be displayed to the user during requirements trace retrieval.

Future work will incorporate the findings of this study into our requirements traceability tool and will include conducting a larger scale empirical study that investigates the broader support provided by our trace retrieval tool for software engineering related activities such as requirements validation and impact analysis.

Acknowledgments

The work described in this paper was partially funded by NSF grant CCR-0306303. We also thank Grace Bedford, an undergraduate HCI student at DePaul University for her invaluable input in incorporating the findings of this study to improve the usability of our trace retrieval tool.

References