

# Human Recoverability Index: A TraceLab Experiment

Alex Dekhtyar

Computer Science Department  
California Polytechnic State University  
San Luis Obispo, CA, United States  
dekhtyar@calpoly.edu

Michael Hilton

Computer Science Department  
California Polytechnic State University  
San Luis Obispo, CA, United States  
hilton@calpoly.edu

**Abstract**—It has been generally accepted that not all trace links in a given requirements traceability matrix are equal - both human analysts and automated methods are good at spotting some links, but have blind spots for some other. One way to choose automated techniques for inclusion in assisted tracing processes (i.e., the tracing processes that combine the expertise of a human analyst and special-purpose tracing software) is to select the techniques that tend to discover more links that are hard for human analysts to observe and establish on their own. This paper proposes a new measure of performance of a tracing method: human recoverability index-based recall. In the presence of knowledge about the difficulty of link recovery by human analysts, this measure rewards methods that are able to recover such links over methods that tend to recover the same links as the human analysts. We describe a TraceLab experiment we designed to evaluate automated trace recovery methods based on this measure and provide a case study of the use of this experiment to profile and evaluate different automated tracing techniques.

## I. INTRODUCTION

TraceLab [6], [1] is a visual experimental workbench built to support design, implementation and execution of traceability experiments. Built at DePaul University for the traceability research community, TraceLab is designed to become the standard way by which empirical studies comparing tracing methods and approaches are conducted.

TraceLab incorporates a large collection of various *modules*: software components that can be used in various automated and semi-automated tracing processes, including components for importation of textual information, text processing routines (such as stopword removal and stemming), information retrieval and text mining techniques, and components that measure the accuracy of candidate traces. TraceLab is designed to be extensible: traceability research groups are encouraged to contribute new modules to the TraceLab library, and several research groups have already made such contributions [5]. In addition, TraceLab provides a visual environment allowing users to create out of existing components, set up, and execute traceability experiments.

In April of 2012, a group of traceability researchers met in Chicago to discuss the plans for using TraceLab for traceability research. Participants of the meeting were asked to design, and, later, to build a traceability experiment using the TraceLab framework.

As part of their participation in the meeting, the authors expressed interest in designing a TraceLab experiment in support of assisted tracing [2]. Some of the prior research by one of the authors, concentrated on studying what human analysts do with the results of automated methods and whether they tend to recognize good advice provided by the automated methods [3], [2], [4]. This work indicated that not all trace links are equally easily recognizable by human analysts: some links present in the experimental studies were almost uniformly recovered and recognized by humans, while other links were extremely difficult for them. From this observation, we have come up with the idea of measuring whether a specific automated method is capturing the links in a fashion *complementary* to the analysts.

This paper presents the TraceLab experiment that evaluates automated tracing methods based on their ability to recover trace links that are *difficult* for humans to recognize and recover. Methods that capture links that are harder for humans to discover are rewarded. Methods that tend to recover the same links as the human analysts are penalized.

The key contributions of this paper are three-fold:

- 1) We introduce a new weighted coverage measure, HRI recall (human recoverability-indexed recall), which favors trace relations that contain links that human analysts have difficulty recognizing.
- 2) We describe our implementation of a TraceLab experiment that compares different automated tracing techniques based on the HRI recall and precision measures. The experiment, in its current form, uses two small datasets for which data on the difficulty of link discovery by human analysts has been collected in prior research.
- 3) We present the results from a pilot run of the experiment using a number of trace techniques currently available in TraceLab. The results suggest that the HRI recall measure can indeed distinguish between techniques based on their tendency to discover harder-to-find (for humans) links.

## II. MEASURING HUMAN RECOVERABILITY OF TRACE LINKS

Consider a pair of textual artifacts of the software engineering process,  $R$  and  $D$ . We refer to document  $R$  as the

high-level document, and to  $D$  as the low-level document, as we consider the scenario of tracing  $R$  to  $D$ . Let  $R$  consist of  $n$  individual elements:  $R = (r_1, \dots, r_n)$ , and  $D$  consist of  $m$  individual elements:  $D = (d_1, \dots, d_m)$ .

We denote as  $T$ , the true traceability relationship between  $R$  and  $D$ :

$$T = \{(r, d) | r \in R, d \in D, r \text{ traces to } d\}.$$

Let  $T = \{\ell_1, \dots, \ell_N\}$ , where  $\ell_i$  denotes a true link  $\ell_i = (r^i, d^i)$ .

Consider a group of  $k$  experts  $A = \{a_1, \dots, a_k\}$ . Each expert  $a_j$  performs an independent trace of  $R$  to  $D$ . Let

$$T_{a_j} = \{\ell_{j1}, \dots, \ell_{jN_j}\}$$

be the traceability relationship (traceability matrix) recovered by expert  $a_j$ .

*Definition 1:* Let  $\ell = (r, d) \in T$  be a true trace link between  $R$  and  $D$ . The *discovery set*  $A(\ell)$  is defined as:

$$A(\ell) = \{a \in A | \ell \in T_a\}$$

Informally, the discovery set of a true trace link is the subset of experts who included the link in their traces.

*Definition 2:* Let  $\ell = (r, d) \in T$  be a true trace link between  $R$  and  $D$ . The *recovery ratio* of  $\ell$  given the set of experts  $A$  is defined as

$$RR(\ell) = \frac{|A(\ell)|}{|A|} = \frac{|A(\ell)|}{k}.$$

The recovery ratio of a trace link is the percentage of experts who discovered it.

Given a group of experts, the recovery ratio of a trace link among that group can be viewed as a measure of difficulty of link discovery. Indeed, we expect that more straightforward, more obvious trace links are discovered by more experts when they perform trace recovery. Links with small recovery ratios are the ones that, for one reason or another, are hard for humans to observe/detect.

Our goal is to develop a measure of "goodness" of a candidate trace relationship that rewards candidate traces that contain harder-to-find links, i.e., links with lower recovery ratios. The next measure, *human recoverability score* essentially turns the recoverability ratio into a reward score: the more difficult it is to discover the link, the higher the reward.

*Definition 3:* Given a true trace link  $\ell = (r, d) \in T$ , and the set of experts  $A$ , the *human recoverability index score (HRI score)* of  $\ell$  w.r.t.  $A$  is

$$HRI\text{Score}(\ell) = 1 - RR(\ell).$$

A link  $\ell$  recovered by all human experts receives the HRI score of 0, meaning that we are essentially not interested in the ability of an automated method to discover this link. Conversely, we give highest priority to the ability of automated methods to detect links with the HRI score of 1: the links not found by any human expert.

Rewards for detecting individual links need to be combined into a single metric of a candidate trace relationship  $T'$ .

*Definition 4:* Let  $H \subseteq T$  be some subset of the ground truth. The *human recoverability score* of  $H$  is defined as

$$\omega(H) = \sum_{\ell \in H} HRI\text{Score}(\ell).$$

In particular,

$$\omega(T) = \sum_{i=1}^N HRI\text{Score}(\ell_i) = N - \sum_{i=1}^N RR(\ell_i).$$

Using the definition of the human recoverability score of a set of true links, we can now define the *human recoverability indexed recall*, our measure of the ability of an automated method to recover hard-to-discover links.

*Definition 5:* Let  $T' = \{\ell'_1, \dots, \ell'_M\}$  be a candidate trace recovered by some automated method  $\mathcal{M}$  when tracing  $R$  to  $D$ . Let  $T' = H' \cup S'$ , where  $H' \subseteq T$  are the true links discovered by  $\mathcal{M}$ , while  $S' \cap T = \emptyset$  are the false positives.

The *human recoverability-indexed recall (HRI recall)* of  $T'$  is defined as

$$HRI\text{Recall}(T') = \frac{\omega(H')}{\omega(T')}.$$

The HRI recall is an extension of the regular recall measure, as shown by the following theorem.

*Theorem 1:* Let  $T = \{\ell_1, \dots, \ell_N\}$  be a trace relation between two artifacts  $R$  and  $D$ , and let  $A = \{a_1, \dots, a_k\}$  be  $k$  human experts who performed a trace recovery task between  $R$  and  $D$ , and retrieved traces  $T_{a_1}, \dots, T_{a_k}$  respectively.

If  $RR(\ell_1) = RR(\ell_2) = \dots = RR(\ell_k)$ , then for any candidate trace relation  $T'$ ,

$$HRI\text{Recall}(T') = \text{Recall}(T').$$

The theorem above states that if all links in the trace relation have exactly the same recovery ratio, i.e., have been discovered at exactly the same rate, then the HRI recall of any candidate trace is equal to its (standard) recall.

### III. TRACELAB EXPERIMENT

Tracelab experiments are composed from a set of executable components and decision nodes, all of which are laid out in the form of a precedence graph on a canvas. Components can be primitive or composite. Composite components can be defined by the user or imported. The TraceLab experiment that we constructed consists of both existing Tracelab components and some custom components that we developed.

*a) Experiment flow.:* The experiment uses a number of specially prepared and annotated datasets and is run on one or more automated tracing methods. The experiment structure is outlined below.

- 1) **Datasets.** One or more datasets are used for the experiment. For the experiment, each dataset  $\mathcal{D}$  consists for the following components:  $\mathcal{D} = (R, D, T)$ , where  $R = \{r_1, \dots, r_n\}$  and  $D = \{d_1, \dots, d_m\}$  are the

high-level and the low-level artifact respectively, and  $T = \{(r^1, d^1, \alpha^1), \dots (r^N, d^N, \alpha^N)\}$  is an annotated true trace relation between  $R$  and  $D$ . Each link  $\ell = (r, d) \in T$  is annotated with a parameter  $\alpha = RR(r, d)$ . That is, each dataset used in the experiment, must have prior data from tracing experiments involving human experts associated with it. Each true link from the trace relation between the high- and the low-level document of the dataset is annotated with its recovery ratio based on the data from the prior tracing experiments. Without loss of generality we assume that the recovery ratios are all constructed based on the same set of human analysts and the candidate trace relations constructed by them during such a prior experiment.

- 2) **Methods tested.** The experiment evaluates one or more "standard" automated tracing method. Each evaluated method must be implemented as a (possibly compound) TraceLab component, which takes as input a pair of textual artifacts (in any format that TraceLab can handle) and outputs a candidate trace relationship (a candidate requirements tracing matrix) as the result of its work.
- 3) **Experiment flow.** The experiment proceeds as follows. For each dataset, each test method is run and the candidate trace is constructed. The candidate trace is evaluated for precision, recall and HRI recall, and the computed measures are reported to the TraceLab users.

Figure 1 depicts the actual architecture of the TraceLab experiment we developed. Below, we discuss the experiment in more detail.

The experiment design consists of three stages. The first stage, from the "Start" meta-component and up to the first component inside the "Tracing Method" box, represents the iteration over multiple datasets used in the experiment. The components for this stage are standard TraceLab modules. The second stage is the collection of components inside the "Tracing Method" box. These components jointly represent the tracing methods used in the experiment. In an experiment comparing submissions from multiple research groups, each method would normally be defined as a composite component. In the case of our pilot study it has been expanded for illustrative purposes. The tracing and the components forming it are examples that come with TraceLab.

The third stage of the experiment is the collection and evaluation of the results. This stage is represented by the three final components in Figure 1. **MetricComputation-Component** (highlighted in Figure 1 is a new component designed for this experiment. It extended TraceLab's standard component for computation of accuracy metrics by adding the computation of HRI Recall, HRI Recall Number Links Curve, HRIRecall Percentage Curve, and Precision HRIRecall Curve.

HRIRecall computes the HRI Recall score for the current dataset. This is analogous to the Recall computation that already exists in TraceLab. In addition to computing the HRI Recall scores, there are some metrics computed about the score, including the mean, and standard deviation.

HRIRecall Number Links Curve graphs the number of links that have been found against the HRI Recall score. This graph shows the HRI Recall against the number of links that are found. In contrast, HRIRecall Percentage Curve has a similar graph, but instead of the number of correct links found, the HRI Recall is graphed against the percentage of correct links found, or in other words, precision.

Precision HRIRecall curve is a graph of the HRI Recall vs the Precision data. It is the same data as HRIRecall Percentage Curve, but with the axis flipped. It is presented in the same format as the Precision Recall Curve, which allows for one to compare the Precision Recall Curve and the Precision HRI Recall Curve for the same result set.

## IV. EVALUATION

### A. Experimental Design

The original intent of community-developed TraceLab experiments was to include them as "competitions", challenging traceability research teams to submit their best techniques to the evaluation by each experiment. The competition aspect of the experiments was later abandoned. For our part, we conducted a proof-of-concept evaluation study of our experiment using a number of tracing methods available in TraceLab.

The purpose of the evaluation study was to test the overall use of the HRI recall measure as the means of evaluating how well automated tracing methods recover hard-to-detect trace links. We discuss the design of the study below.

**Datasets:** We used two datasets in this study. Both datasets, Cal Poly[3] and WARC[8] have been used in prior studies of human analyst behavior in assisted tracing scenarios. Each dataset is briefly described below.

**Cal Poly dataset:** Cal Poly dataset, otherwise known as **ChangeStyle** consists of 32 requirements and 17 system tests for a BlueJ plugin Java code formatting tool named **ChangeStyle**. This software was a project for one of Cal Poly's junior Software Engineering two-quarter course sequences. It has been used in a number of prior human analyst studies [3], [2], [4], [7]. The true traceability matrix for this dataset contains 23 links. While different studies were conducted in different ways, a total of 84 candidate traceability matrices produced by human analysts in the experiments described in [3], [2], [4], [7] were available to us. All were used to compute the recovery ratios for individual trace links. Figure 2 shows the diagram of the HRI scores for the dataset. Each bar depicts the HRI score of a single true link, the links are presented in ascending order of the HRI scores. As seen from the figure, the human recoverability index ranges from almost 0 for a pair of links that were discovered by almost all human analysts, to one link with an HRI score of 0.946, which was discovered by almost noone. Overall, there is a significant difference in recoverability scores of individual links in this dataset. Table I shows some of the statistics (smallest, largest, average, standard deviation and median) for the HRI scores for this dataset and the WARC dataset discussed below.

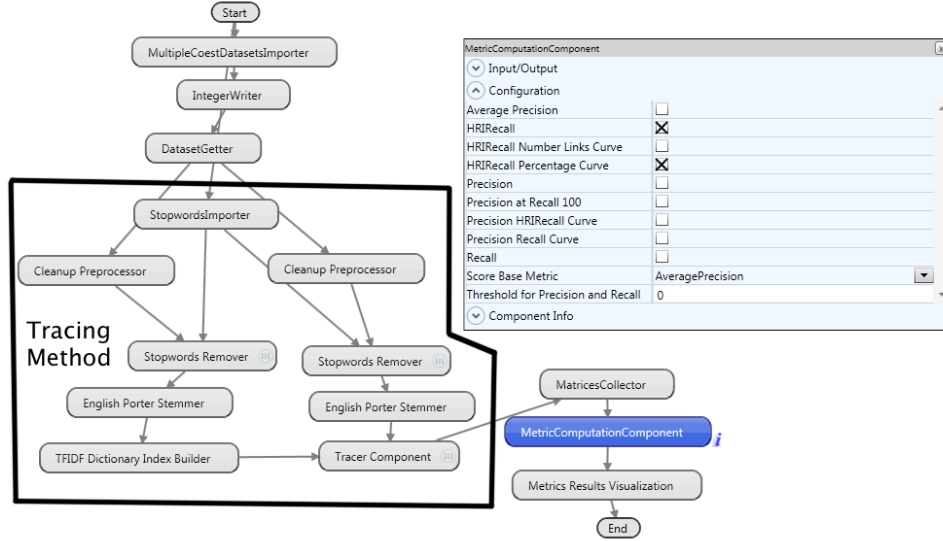


Fig. 1. Tracelab Experiment

**WARC dataset:** The WARC dataset consists of 42 functional requirements and 89 software requirements for an open source suite of web archive file manipulation tools<sup>1</sup>[8]. The true trace for this dataset consists of 55 links. In [8], a tracing experiment using this dataset involved 24 human experts. As a result, we had access to 24 candidate traceability matrices submitted by human analysts in [8]. The distribution of the HRI scores for this dataset is shown in Figure 3. Two links have HRI scores of 0 (found by everyone), while on the other end of the spectrum, one link has a score of 0.958 (almost never found). This distribution shows that there are some relatively easy to recognize links in the WARC dataset, along with a few links that were very hard for most people to detect.

TABLE I  
HRI SCORE STATISTICS FOR THE DATASETS USED IN THE PILOT STUDY.

	WARC	CP
MAX	0.958	0.946
Min	0	0.027
Avg	0.335	0.573
Std Dev	0.260	0.252
Median	0.292	0.622

**Tracing Methods used:** We used the default tracing component supplied by TraceLab as the collection of tracing methods for the experiment. The default component implements Vector Space Retrieval and uses four different techniques for computing the similarity between two textual elements. The four computation methods use *simple matching*, *Jaccard*, *Dice* and *cosine* similarity scores. Each of the four similarity computations was treated as a separate tracing method for the purpose of this study.

<sup>1</sup><http://code.google.com/p/warc-tools>

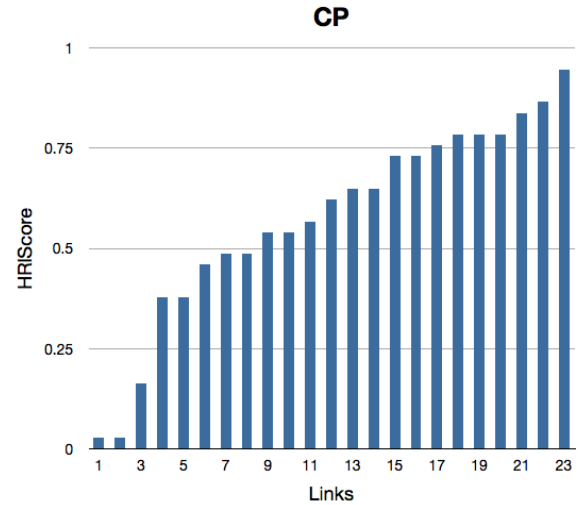


Fig. 2. HRI Score Distribution for all true links in Cal Poly Dataset

**Measures collected.:** Each method was run on each of the datasets. The result was a list of candidate links, with a similarity score associated with each candidate link. We collected three measures: precision, recall and HRI recall. Each list of candidate links is ordered in descending order by the similarity score of the links. We then use *recall* as an independent variable, and compute the other two measures: *precision* and *HRI recall* starting at the recall level closest to 10%, proceeding in about 10% increments until we reach the last true link reported by each method. The results are plotted in three ways:

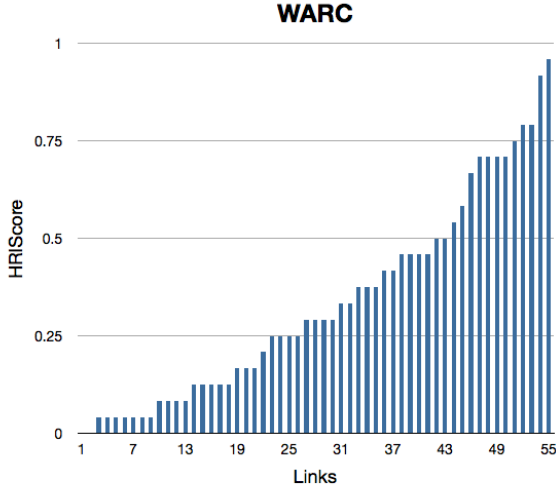


Fig. 3. HRIScore Distribution for all true links in WARC Dataset

- (a) *precision vs. recall* to provide the baseline for observations;
- (b) *precision vs. HRI recall* to compare to the precision vs. recall graphs, and;
- (c) *HRI recall vs. recall* to look at the impact of using HRI scores as weights of recovered links.

We report results in both graphical and tabular form.

## B. Results

The results of the pilot study are presented in Figures 4, 5, 6, 7, 8 and 9 and in Tables II, III, IV and V.

For CalPoly dataset, all trace recovery methods reached just under 80% recall. However, as seen from the HRI recall plots, all methods except for (surprisingly!) *simple match* have mostly been recovering the easy-to-find trace links. As Figure 9 shows, *simple match* method captures more difficult-to-find links, especially, at the beginning of the candidate link list for this method.

For the WARC dataset all methods were roughly equally unsuccessful in capturing a significant number of true links - reaching only recall of around 30%. Most of the recovered links, as evidenced from Figures 7 and 8 and Table V were with high recovery ratios (easy to find). Simple matching method, again recovered links that were somewhat different and had higher HRI scores.

The pilot study shows that even when in the case of relatively straightforward tracing methods, it is possible to evaluate different methods on the basis of their ability to retrieve harder-to-find links. The overall HRI recall, and HRI recall vs. recall plot can be used to determine how different methods behave.

## V. DISCUSSION

There are different ways in which one could evaluate automated methods for appropriateness of their use in assisted tracing tasks. The ability of an automated method to detect

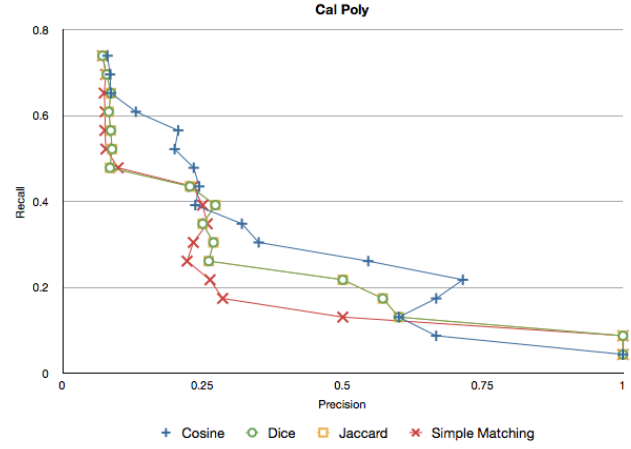


Fig. 4. Cal Poly Dataset: precision vs. recall

TABLE II  
CAL POLY DATASET: PRECISION VS. RECALL.

Recall	Precision			
	SimpleMatching	Jaccard	Cosine	Dice
10%	0.5	0.6	0.6	0.6
20%	0.263	0.5	0.714	0.5
30%	0.233	0.269	0.35	0.269
40%	0.232	0.227	0.243	0.227
50%	0.077	0.088	0.2	0.088
60%	0.076	0.082	0.131	0.082
70%	0.071	0.071	0.080	0.071

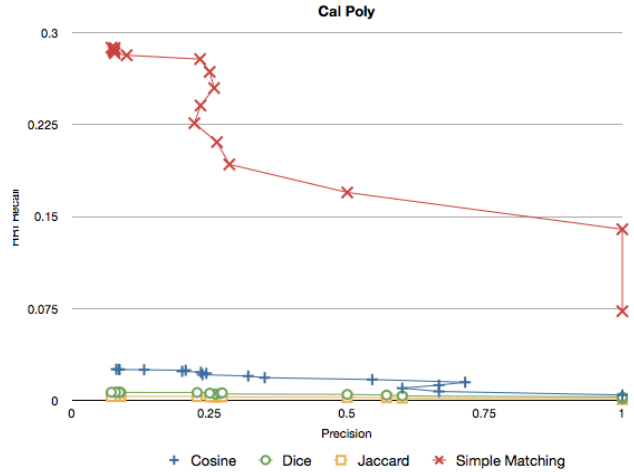


Fig. 5. Cal Poly Dataset: precision vs. HRI recall

links that human analysts tend to miss is one possible measure of how useful the method can be.

In this paper we presented the notion of measuring the quality of a candidate trace in a way that rewards the presence of hard-to-detect links and discounts the presence of links that are easily recoverable by humans. We formally defined the human recoverability index score of a link and the human recoverability-indexed recall of a candidate trace relation. Using two datasets from prior studies of assisted tracing we

TABLE III  
CAL POLY DATASET: PRECISION VS. HRI RECALL

SimpleMatching		Jaccard		Cosine		Dice	
HRI Recall	Precision	HRI Recall	Precision	HRI Recall	Precision	HRI Recall	Precision
0.17	0.5	0.002	0.6	0.01	0.6	0.004	0.6
0.211	0.263	0.002	0.5	0.015	0.714	0.005	0.5
0.241	0.233	0.003	0.269	0.018	0.35	0.006	0.269
0.278	0.233	0.003	0.227	0.022	0.244	0.006	0.227
0.283	0.077	0.003	0.088	0.024	0.2	0.007	0.088
0.286	0.076	0.003	0.082	0.025	0.131	0.007	0.082
0.288	0.071	0.003	0.071	0.025	0.08	0.007	0.071

TABLE IV  
WARC DATASET: PRECISION VS. RECALL

SimpleMatching		Jaccard		Cosine		Dice	
HRI Recall	Precision	HRI Recall	Precision	HRI Recall	Precision	HRI Recall	Precision
0.161	0.071	0.008	0.073	0.025	0.1	0.015	0.073
0.236	0.0710	0.0104	0.081	0.036	0.097	0.020	0.081
0.263	0.049	0.011	0.057	0.040	0.048	0.022	0.057

TABLE V  
WARC DATASET: PRECISION VS. HRI RECALL

HRI recall	Precision			
	SimpleMatching	Jaccard	Cosine	Dice
10%	0.054	0.0597	0.0905	0.0605
20%	0.058	0.064	0.081	0.065
30%	0.057	0.062	0.072	0.062

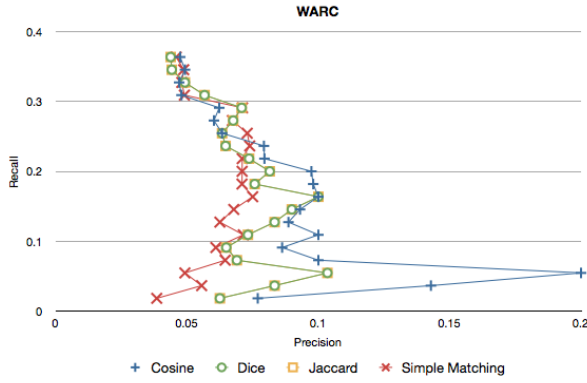


Fig. 6. WARC Dataset: precision vs. recall

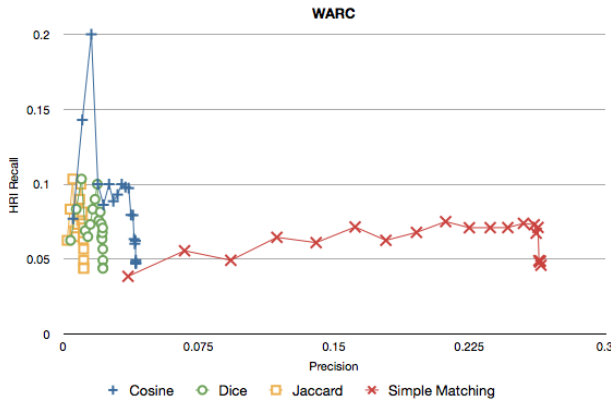


Fig. 7. WARC Dataset: precision vs. HRI Recall

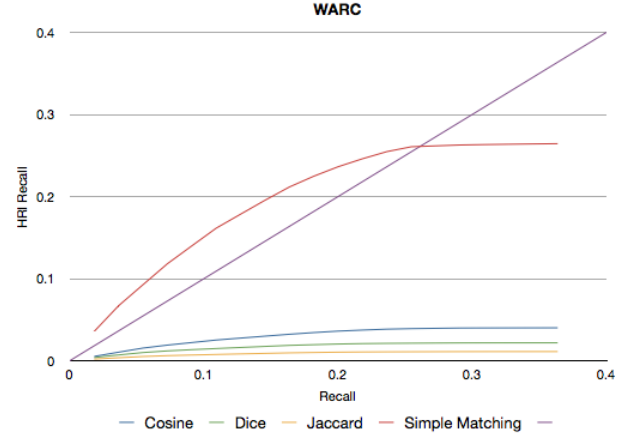


Fig. 8. WARC dataset: HRI recall vs recall

constructed a TraceLab experiment that measures HRI recall of candidate traces returned by various automated tracing methods and conducted a small pilot study. The new and modified TraceLab components and the experiment description

are being made available to the traceability community.

The proposed measure and experiment are not without some weak spots. In particular, the HRI recall measure as presented and used in this paper completely discounts the ability of an automated method to recover trace links that have a 100% recoverability ratio. This means that on datasets with significant numbers of such links, methods that ignore those links (and thus have lower overall recall) receive the same HRI recall scores as methods that recover the same links with recoverability ratios below 100%, but *in addition* also recover the trace links with 100% *RR* (and thus have higher overall recall). This is a bit counterintuitive, so, a version of HRI recall

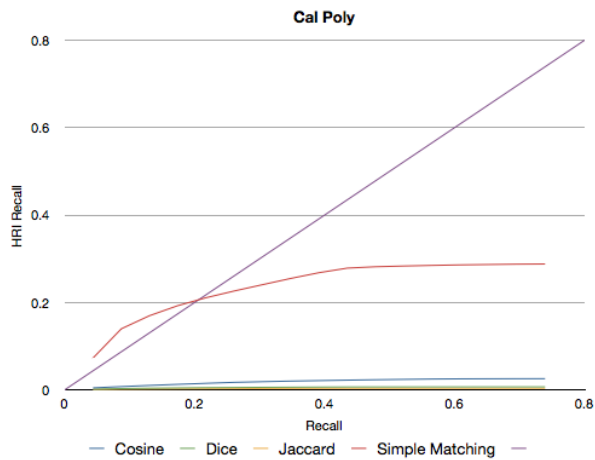


Fig. 9. CalPoly dataset: HRI recall vs recall

that establishes an *ambient* minimal contribution for each link may be in order.

Another challenge in using the proposed framework comes from the relative scarcity of datasets usable for the experiments. Unlike traditional traceability experiments, just having a true traceability matrix for the dataset is not sufficient. We need to know the recoverability ratios for each link in the true trace, which means that the dataset had to have been traced by multiple human analysts. In this paper, we used two datasets that have been traced by multiple human analysts in a series of prior experiments [3], [2], [4], [7], [8]. As assisted tracing is studied in more and more detail, we hope to see other datasets used in multiple human-centered experiments.

Nevertheless, we believe that the use of HRI recall and similar measures can help us determine automated tracing methods that can be used in assisted tracing scenarios to complement the expertise of human experts.

#### ACKNOWLEDGMENTS

The authors would like to thank Jane Cleland-Huang for leadership in developing TraceLab and for bringing us to

Chicago, the TraceLab development team, especially Adam Czauderna for helping us debug TraceLab code, Wei-Keat Kong and Jane Hayes for providing access to the internals of WARC dataset and John Dalbey for developing the Cal Poly dataset.

#### REFERENCES

- [1] J. Cleland-Huang, A. Czauderna, A. Dekhtyar, O. Gotel, J. Hayes, E. Keenan, G. Leach, J. Maletic, D. Poshyvanyk, Y. Shin, A. Zisman, G. Antoniol, B. Berenbach, and P. Maeder. Grand challenges, benchmarks, and tracelab: Developing infrastructure for the software traceability research community. In *Proceedings, 6th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE2011)*, May 2011.
- [2] D. Cuddeback, A. Dekhtyar, J. Hayes, J. Holden, and W.-K. Kong. Towards overcoming human analyst fallibility in the requirements tracing process (nir track). In *Proceedings, International Conference on Software Engineering (ICSE2011)*, pages 860–863, May 2011.
- [3] D. Cuddeback, A. Dekhtyar, and J. H. Hayes. Automated requirements traceability: The study of human analysts. In *Proceedings, 18th International Requirements Engineering Conference (RE'2010)*, pages 231–240, 2010.
- [4] A. Dekhtyar, O. Dekhtyar, J. Holden, J. Hayes, D. Cuddeback, and W.-K. Kong. On human analyst performance in assisted requirements tracing: Statistical analysis. In *Proc., 19th International Requirements Engineering Conference (RE2011)*, pages 111–120, September 2011.
- [5] B. Dit, E. Moritz, and D. Poshyvanyk. A tracelab-based solution for creating, conducting, and sharing feature location experiments. In *Proceedings, International Conference on Program Comprehension (ICPC'12)*, pages 203–208, 2012.
- [6] E. Keenan, A. Czauderna, G. Leach, J. Cleland-Huang, Y. Shin, E. Moritz, M. Gethers, D. Poshyvanyk, J. Maletic, J. Huffman Hayes, A. Dekhtyar, D. Manukian, S. Hossein, and D. Hearn. Tracelab: an experimental workbench for equipping researchers to innovate, synthesize, and comparatively evaluate traceability solutions. In *Proceedings of the 2012 International Conference on Software Engineering, ICSE 2012*, pages 1375–1378, Piscataway, NJ, USA, 2012. IEEE Press.
- [7] W.-K. Kong, J. Hayes, A. Dekhtyar, and J. Holden. How do we trace requirements? an initial study of analyst behavior in trace validation tasks. In *Proceedings, 4th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE'2011)*, May 2011.
- [8] W.-K. Kong, J. H. Hayes, A. Dekhtyar, and O. Dekhtyar. Process improvement for traceability: A study of human fallibility. In *Proc. 20th IEEE International Requirements Engineering Conference (RE2012)*, pages 31–40, September 2012.