Empir Software Eng (2010) 15:119–146
DOI 10.1007/s10664-009-9114-z

Improving automated requirements trace retrieval: a study of term-based enhancement methods

Xuchang Zou · Raffaella Settimi · Jane Cleland-Huang

Published online: 28 July 2009

Editor: Daniela Damian

Abstract Automated requirements traceability methods that utilize Information Retrieval (IR) methods to generate and maintain traceability links are often more efficient than traditional manual approaches, however the traces they generate are imprecise and significant human effort is needed to evaluate and filter the results. This paper investigates and compares three term-based enhancement methods that are designed to improve the performance of a probabilistic automated tracing tool. Empirical studies show that the enhancement methods can be effective in increasing the accuracy of the retrieved traces; however the effectiveness of each method varies according to specific project characteristics. The analysis of such characteristics has lead to the development of two new project-level metrics which can be used to predict the effectiveness of each enhancement method for a given data set. A procedure to automatically extract critical keywords and phrases from a set of traceable artifacts is also presented to enhance the automated trace retrieval algorithm. The procedure is tested on two new datasets.

Keywords Requirements traceability · Requirements management · Information retrieval models.

1 Introduction

Requirements traceability, which is defined as "*the ability to describe and follow the life of a requirement, in both a forwards and backwards direction*" (Gotel and Finkelstein 1994), is concerned with managing the relationships between requirements and other

X. Zou · R. Settimi (✉)
School of Computing, DePaul University, Chicago, IL, USA
e-mail: rsettimi@cdm.depaul.edu

X. Zou
e-mail: xzou@cdm.depaul.edu

J. Cleland-Huang
System and Requirements Engineering Center, School of Computing, DePaul University, Chicago, IL, USA
e-mail: jhuang@cdm.depaul.edu

artifacts developed as part of the software development lifecycle. Traceability provides critical support to facilitate a broad range of software development tasks including requirements validation, impact analysis, and compliance verification. As such, the development of effective traceability strategies has been widely recognized as an important activity.

A variety of commercial tools and research prototypes have been created to support traceability tasks. These approaches employ varied techniques such as cross referencing (Evans 1989), use of traceability matrices (Davis 1990), hypertext (Kaindl 1993), and templates (IDE 1991). However, the biggest drawback of these tools, is the intensive effort that they demand from the analyst to create and maintain links. For instance, most tools which use traceability matrices to map and record the relationships between two corresponding types of artifacts, require the analyst to manually construct the matrix and maintain links to accurately reflect evolutionary changes in the system. Practice has repeatedly shown that this imposes a huge burden on the analyst especially when applied to large and complex systems.

In recent years, Information Retrieval (IR) techniques have been used successfully to dynamically generate and retrieve traceability links on an "as-needed" basis (Antoniol et al. 2000; Hayes et al. 2006; Maletic et al. 2003; Marcus and Maletic 2003; Settimi et al. 2004; Cleland-Huang et al. 2005a; Cleland-Huang et al. 2005b; De Lucia et al. 2007). Because software artifacts such as requirements, design documents and source code contain large amounts of textual information, IR techniques can be used to evaluate their textual similarity and generate relevant traces. IR-based approaches eliminate the upfront effort of establishing and maintaining a traditional traceability infrastructure such as a matrix or a set of hyperlinks, and several recent studies have demonstrated the effectiveness of IR-based automated traceability tools to help construct and maintain traceability matrices (Antoniol et al. 2000, Hayes et al. 2006; Maletic et al. 2003; Marcus and Maletic 2003; Settimi et al. 2004; Cleland-Huang et al. 2005a; Cleland-Huang et al. 2005b).

To be effective, a tracing tool must retrieve as many correct traceability links as possible. Unlike a typical search tool, in which precision is favored over recall, the traceability problem dictates that traceability tools must retrieve a high percentage of targeted traceability links in order to be useful. The two standard IR metrics of *recall* and *precision* are generally used to assess the effectiveness of requirements tracing tools where *recall* is defined as the proportion of correct links that are retrieved by the tool over all relevant traceability links, and *precision* is the proportion of correct links over all retrieved links (Frakes and Baeza-Yates 1992). To evaluate the distribution of correct traces within the list of retrieved links more effectively, a new metric named *Average Precision Change* is presented in Section 3.1.

Previous empirical studies analyzing the efficiency of automated trace retrieval methods indicate that at high recall levels of 90%, precision is usually below 40% and sometimes even less than 10% (Antoniol et al. 2000; Hayes et al. 2006; Maletic et al. 2003; Marcus and Maletic 2003; Settimi et al. 2004; Cleland-Huang et al. 2005a; Cleland-Huang et al. 2005b). This low precision indicates that many false traces are incorrectly retrieved by the tool and consequently the user needs to manually evaluate a long list of retrieved links in order to identify the correct traces. Despite this precision problem, automated IR approaches still significantly reduce the effort required by traditional traceability methods to manually evaluate traces. In our experiments, for instance, the list of candidate traces retrieved by our automated tracing tool at recall levels of 90%, contains on average less than 30% of all possible links. In other words instead of checking all potential traces, the analyst's work is reduced to checking a much

smaller fraction of documents. A study by (De Lucia et al. 2009) confirmed that automated traceability tools significantly reduce the time a software engineer spends evaluating links, and also improve the accuracy of his or her evaluation. However, improving the precision of retrieved traces can help increase the user's confidence in the accuracy of the tool and may facilitate the adoption of IR-based automated traceability methods in industry.

Prior research in automated trace retrieval has attempted to improve precision in several different ways. Some approaches have adopted standard IR techniques while others have incorporated specific characteristics of the software artifacts collection to improve retrieval algorithms. This prior work has included using a thesaurus to capture synonyms or expand acronyms in both the query and the traced artifacts in order to mitigate the problem in which two related artifacts have no matching terms (Hayes et al. 2006; Lin et al. 2006); utilizing hierarchical structure information such as headings in requirements documents or package names for class diagrams to enhance the relevance score between a query and a traced artifact if their ancestor documents share the same terms (Cleland-Huang et al. 2005b); and *user relevance feedback* in which potential traces are evaluated by the user and term weights are adjusted depending on whether the terms belong to relevant or irrelevant traces on the basis of the information entered by the user (Rocchio 1971; Hayes et al. 2003).

Although delivering some improvements in precision, each of the methods has its own drawbacks. For example, some methods, such as the hierarchical approach, are only applicable to datasets which exhibit specific characteristics such a strong internal hierarchy, while other techniques such as user relevance feedback require additional effort from the analyst in order to improve results. To further increase precision of the results, an analysis was conducted of the trace retrieval results. As a result, common patterns were identified among some of the links that were incorrectly handled by the tracing tool. This in turn led to the development of three effective enhancement strategies that are simple, easy to implement and require minimal additional effort from the analyst's side.

1. *Query Term Coverage* (TC) is an enhancement strategy designed to increase the relevance ranking of traces between software artifacts that have more than one unique word in common (Zou et al. 2007). This approach increases the accuracy of the trace results by reducing the number of incorrectly retrieved links between unrelated pairs of software artifacts that contain a single matching word which co-occurs multiple times.
2. *Phrasing* is designed to improve the retrieval precision by identifying phrases and increasing the similarity scores for pairs of artifacts that contain shared phrases (Zou et al 2006).
3. The *Project Glossary* approach utilizes the project glossary's content to identify critical terms and phrases that should be weighted more heavily than others, as they can be regarded more meaningful in identifying traceability links. A keyword extraction method is proposed to dynamically discover critical keywords and sentences from the set of traceable artifacts for a given project. (Zou et al. 2006; Zou et al. 2008).

This paper presents results of a new empirical study that evaluates and compares the effectiveness of these three enhancement strategies, using datasets containing a variety of software systems including research projects, student assignments, and industrial applications. The study shows that the effectiveness of the enhancement strategies correlates with certain textual characteristics of the software artifacts measured using the two new metrics of *average query term coverage* (QTC) and *average phrasal term coverage* (PTC). These metrics are used to evaluate the performance of the proposed

approaches in improving the trace retrieval accuracy. The main contributions of this paper can therefore be summarized as follows:

- The introduction and evaluation of three enhancement strategies and new synergistic approaches implementing the enhancement strategies concurrently. New datasets are used in the reported experiments and results extend previously published studies for both the term coverage and the phrasing approaches.
- Three new metrics are defined. *Average Precision Change (AP)* defined in Section 3.1 is an effective measure to compare the impact of different approaches on automated trace retrieval. The A*verage* Q*uery Term Coverage* (QTC) and A*verage* P*hrasal Term Coverage* (PTC) are proposed in Section 4.1 to predict the effectiveness of the Query Term Coverage and the Phrasing approaches respectively in improving the tracing results accuracy.
- A new iterative algorithm is presented to identify which enhancement strategy may be effective in improving the accuracy of automated tracing results for a specific project. The algorithm starts with an initial step that uses the predictive metrics QTC and PTC to select an enhanced tracing strategy. Subsequent steps refine the initial enhancement decisions by collecting and evaluating user feedback in real-time. A case study is presented to illustrate the application of the iterative approach.
- New experiments involving two new datasets are discussed in Section 5.2 to evaluate the effectiveness of a keyword extraction method that identifies critical terms and phrases from the collection of artifacts to be traced. Such terms can be used in the Project Glossary technique to improve the accuracy of tracing tools.

The paper provides a detailed description of the three enhancement strategies, *Query Term Coverage (TC)*, *Phrasing* and use of a *Project Glossary* in Section 2, and then Section 3 reports experimental results comparing these three strategies. Factors that impact the effectiveness of the three proposed strategies are investigated in Section 4, and new metrics are defined for predicting how well a certain method may perform on a specific dataset. These metrics can be used to construct intelligent tracing tools for automatically determining which enhancement strategy or strategies should be applied to achieve the best retrieval results. Section 5 describes a technique for automatically extracting important keywords and phrases from project requirements. These can be used in place of a missing or incomplete project glossary, and furthermore can be used even when a user defined glossary is present as they tend to include a significant percentage of formal glossary terms. Threats to results validity are discussed in Section 6. Finally, conclusions and future work are summarized in Section 7.

2 Related Work

The three IR models that have been applied most frequently in traceability research studies are the Vector Space Model (VSM) (Salton et al. 1975), the Latent Semantic Indexing (LSI) model (Deerwester et al. 1990), and the Probabilistic Network (PN) model (Wong and Yao 1991). None of these IR models has been proved to be consistently better than the others when applied to requirements trace retrieval.

Although using different approaches, both the VSM and the PN model represent each software artifact as a vector in the space of terms extracted from the entire set of artifacts after some *standard preprocessing steps* that include i) the removal of common words such

as articles, pronouns and conjunctions; and ii) the stemming of words to their root forms by eliminating suffixes and prefixes.

Terms are weighted according to a weighting scheme known as tf-idf (Salton and Buckley 1988) that assigns a relatively high weight to terms that occur many times in an artifact; appear in a small number of artifacts, or both. The two models assign a value to each pair of traceable artifacts known as a *relevance or similarity score* that is computed as a function of the frequency of the terms co-occurring in the two artifacts. Higher relevance scores indicate that artifacts contain the same terms and are therefore potentially related to each other. Only pairs of artifacts that score above a certain value, referred to as the *threshold*, are returned to the analyst for evaluation as potential traceability links. These links are generally displayed in decreasing order of their score.

The LSI approach (Forsythe, 1977), which is based on concept matching, is designed to improve word-matching techniques such as those found in VSM and PN models. The assumption of this model is that there are some underlying semantic structures (latent structures) between documents and terms which are often hidden behind the choice of different words. With LSI, traces between artifacts that contain no shared terms, which would therefore be missed by the standard tf-idf approach, may still be retrieved. In general IR applications, LSI has been shown to perform better on projects with a larger number of artifacts and terms (Furnas et al. 1988; Deerwester et al. 1990).

2.1 A Probabilistic Network (PN) Model

All of the strategies discussed in this paper are implemented as enhancements to the probabilistic network model (Cleland-Huang et al. 2005b). This model computes the relevance score between two artifacts q and d as a conditional probability value $p(d|q)$ defined as a function of the frequency of terms co-occurring in both q and d. The value p(d|q) is computed as follows:

$$p(d|q) = \frac{p(d,q)}{p(q)} = \frac{\sum_{i=1}^{k} p(d|t_i) \times p(q,t_i)}{p(q)} \tag{2.1}$$

where k is the total number of terms extracted from the entire set of traceable artifacts using the preprocessing steps described above.

The three components in the formula follow the *tf-idf* standard weighting strategy. The first component $p(d|t_i) = \frac{freq(d,t_i)}{\sum_{i=1}^{k} freq(d,t_i)}$ represents the relative frequency of term t_i in artifact d and increases with $freq(d, t_i)$, the number of occurrences of t_i in d. The second component $p(q,t_i)$ represents the *inverse document frequency (idf)* and is computed as $p(q,t_i) = \frac{freq(q,t_i)}{n_i}$ where n_i is equal to the number of searchable artifacts $\{d_1, d_2, ..., d_n\}$ containing t_i. It decreases for commonly used terms, reducing their contribution to the overall probability value. The third component $p(q) = \sum_{i=1}^{k} p(q,t_i)$ can be regarded as a scaling factor. Notice that $p(d|q)$ is equal to zero if d and q do not have any terms in common.

The probabilistic automated traceability tool identifies potential traces for a given artifact q, regarded as a *query*, by computing the conditional probability $p(d_j|q_i)$ in expression (2.1) for each traceable artifact d_j in the set $\{d_1, d_2, ..., d_n\}$. Traces for a given query q_i are established by ranking all the artifacts d_j in decreasing order according to the probability scores $p(d_j|q_i)$,

and retrieving as potential traces those pairs (d_j, q) whose probability values $p(d_j|q_i)$ are above a threshold set by the analyst.

Numerous studies in Information Retrieval have suggested that the standard *tf-idf* method used in the probabilistic trace retrieval model can be improved by considering specific characteristics of the documents collection (Jones and van Rijsbergen 1976, Singhal et al. 1999).

An initial study of the retrieval results of the probabilistic tracing tool revealed specific patterns among incorrectly retrieved traces as well as incorrectly missed ones (Zou, 2007). A large percentage of incorrectly retrieved traces were assigned high relevance scores because one or two relatively common terms appeared multiple times in both artifacts. For instance in the Ice-Breaker System (IBS) project describing the requirements and design of a public-works department system for managing roads de-icing (Robertson and Robertson, 1999), the class "*Tutorial GUI*" was incorrectly returned as a trace to the requirement "*A road section shall be added*", even though the two artifacts shared only the term "*section*", occurring very frequently throughout the class. It is also interesting to notice that "*section*" refers to a tutorial section in the class diagram and represents a different and unrelated concept (*road section*) in the requirement.

Furthermore, the analysis showed that traces between artifacts that shared a relatively high number of distinct terms were often missed if the terms had low weight scores and occurred only once in each of the artifacts. This is often the case of project-related terms that are potentially very meaningful but also appear quite frequently in the artifacts. For instance, in the IBS system, a trace between the requirement "*Inventory shall be updated on receipt of a shipment*" and the UML class "*Material Inventory Database*" received a low probability score and was incorrectly rejected by the tool because both of the shared terms "*inventory*" and "*update*" had very low weights according to the *tf-idf* weighting scheme. In summary the study showed that by focusing on matching based on single terms, the standard *tf-idf* term weighting approach misses traces between two artifacts that share a phrase or a set of two or more terms that occur frequently in the artifacts. The next sections present three term-based enhancement strategies of the standard *tf-idf* term weighting schemes that aim to improve the accuracy of automated requirements traceability tools.

2.2 Query Term Coverage (TC) as an Enhancement Factor

The Term Coverage (TC) enhancement method was designed to increase the probability score of traces between a pair of artifacts q and d that share two or more distinct terms. The TC approach defines a new probability score $p_{TC}(d|q)$ between q and d as follows:

$$p_{TC}(d|q) = \begin{cases} m \times p(d|q) & \text{if } p(d|q) < 1/m \\ 1 & \text{if } p(d|q) \geq 1/m \end{cases} \tag{2.2}$$

where $p(d|q)$ is the probability generated from the basic, single-word-matching PN model defined in expression (2.1), and m is the number of distinct terms co-occurring in q and d. The expression (2.2) assigns larger probability scores to artifact pairs that share two or more distinct terms, with an increase that is proportional to the number of distinct matching terms m.

A study analyzing the potential effects of the TC approach on the retrieval results for the IBS project suggests that several missed links are more effectively retrieved when term coverage is applied (Zou et al. 2007). The study measures the *Query Term Coverage*

value, *TC(q,d),* for traces between artifacts q and d in the IBS project. *TC(q,d)* is defined as the proportion of distinct terms in q that are found in a traceable artifact d and is computed as follows:

$$TC(q,d) = \frac{m}{t} \tag{2.3}$$

where t is the number of distinct terms contained in q, and m is the number of distinct matching terms found in both q and d (Burke et al., 1997). Thus a *TC(q,d)* value closer to one indicates that artifacts q and d share a large proportion of distinct terms and therefore are more likely to be conceptually related.

Summary statistics for *TC(q,d)* values computed for all retrieved traces in the IBS project are displayed in Table 1. The average query term coverage is computed for the set of all correctly identified traces (true positives), and all incorrectly retrieved traces (false positives), as well as for the top 100 false positives and the top 100 true positives.

Statistical tests show that the average TC value of false positives is significantly lower than the average TC value for true positives at 5% significance level, and that the average TC value of the top 100 false positives is also significantly lower than the average TC value for top 100 true positives at 5% significance level. Thus the term coverage approach is more likely to increase the probability values for true positives, and to place more correct traces among the top retrieved links in an ordered list of candidate links, thereby increasing the accuracy of the top retrieval results. The improvement is expected to be more significant amongst the top-ranked retrieved links, which will cause more correct links to be returned first to the analyst for inspection.

2.3 Phrasing in Automated Trace Retrieval

The second enhancement strategy of phrasing is designed to improve the retrieval precision by considering phrases that co-occur in pairs of artifacts (Zou et al 2006). Traditional *tf-idf* IR models, such as the VSM and PN models only consider the co-occurrence of single words. However a single word can have multiple meanings or can be related to more than one concept, meaning that a tracing algorithm based only on single word matching may retrieve many irrelevant traces and consequently reduce the precision of the results. Compared to single words, phrases are considered more accurate in capturing the underlying concepts of an artifact. Consider the example for the IBS dataset that was discussed earlier in Section 2.1, in which the basic PN model incorrectly linked the requirement "*A road section shall be added*" to the class "*Tutorial GUI*" because of the single matching term "*section*". However, if phrase matching had been used, the phrases "*road section*" and "*tutorial section*" would not have been matched and the link would not have been retrieved. The use of phrases in automated trace retrieval is intended to reduce the number of false positives and therefore improve the precision of the retrieval results.

Table 1 Comparison on the query term coverage in false positives & true positives in IBS

Test	Group Type	# of links	Average query term coverage	Standard Deviation
1	Top 100 false positives	100	0.298	0.121
	Top 100 true positives	100	0.481	0.222
2	All false positives	1252	0.196	0.103
	All true positives	378	0.310	0.23

Several approaches for phrasing have been proposed in the general IR area (Salton et al. 1974; Fagan 1987; Church and Hanks 1990; Croft et al. 1991). The phrasing approach used in our experiments focuses on two-word phrases defined as sequences of two nouns related through modifications. Previous studies have shown that two-noun phrases are more effective in improving retrieval accuracy (Gay and Croft 1990). Phrases are defined as i) noun-noun phrases such as "*weather forecast*"; and ii) phrases consisting of a noun modified by a prepositional noun, such as "*condition of road*", which can be identified and re-constructed as the noun-noun phrase "*road condition*".

Phrases are identified using a freely available parser-based part-of-speech (POS) tagger named *Qtag* (Tufis and Mason 1998), that determines the syntactic category of each term in the text and outputs POS tags representing the grammatical classes, such as nouns, verbs and adjectives for each term. Qtag is known to provide an accurate and flexible approach for detecting a variety of phrase types that can be easily incorporated into a traceability tool. Qtag is able to retrieve two-noun phrases from artifacts, such as requirements, that contain complete sentences. However, words cannot be tagged in other terser artifacts, such as UML or code methods and class names, as they do not contain complete sentences. As our experiments all involved tracing between requirements and other artifacts, Qtag was used against the requirements in order to identify all two-noun phrases.

The phrasing approach assigns higher scores to traces between artifacts q and d that contain the same phrases. Let S_{PH} be the set of terms contained in phrases found in a requirement q. If no phrases are found, S_{PH} is empty. The phrasing relevance score $p_{PH}(d|q)$ between q and d is computed as follows:

$$p_{PH}(d|q) = p(d|q) + p_f(d|q) \text{ with } p_f(d|q) = \frac{\sum_{t_i \in S_{PH}} p_f(d|t_i) p_f(q, t_i)}{p(q)} \tag{2.4}$$

The new probability score is defined as the sum of two parts. The first part is the basic probability value $p(d|q)$ defined in expression (2.1) that depends solely upon the occurrence of single terms. The second part $p_f(d|q)$ represents the contribution to the probability value provided by phrasing, and depends upon the frequency of terms contained in the identified phrases. This component is equal to zero if artifacts q and d have no phrases in common.

A natural extension to the first two enhancement strategies is to investigate a synergistic implementation of both TC and phrasing into the probabilistic retrieval model. The synergistic approach applies phrasing first, followed by the TC method. The new enhanced probability $p_{TCPH}(d|q)$ between a query q and a document d that incorporates both approaches is calculated similarly to expression (2.2) except that $p(d|q)$ is replaced by $p_{PH}(d|q)$. The expression is defined as follows:

$$p_{TCPH}(d|q) = \begin{cases} m \times p_{PH}(d|q) & \text{if } p_{PH}(d|q) < 1/m \\ 1 & \text{if } p_{PH}(d|q) \geq 1/m \end{cases} \tag{2.5}$$

2.4 Utilizing a Project Glossary

Both the basic probabilistic and the phrasing approaches weigh terms or phrases co-occurring in two artifacts in terms of their frequency of occurrence in the set of artifacts; however they do not take into account the ability of certain phrases and terms to capture more critical concepts for a given project. As generally accepted software engineering

practices define important terms and phrases in a project glossary, our previous work (Zou et al. 2008) proposed a probabilistic retrieval approach that utilized the entries in a project glossary to identify items that should be weighted more heavily than others. Furthermore a glossary may also contain additional phrases that the syntactic parser was unable to discover simply because they did not fit into the prescribed grammatical template.

The automated retrieval approach incorporating project glossary information assigns a higher probability score $p_{PG}(d|q)$ to a trace between artifacts q and d that share either terms or phrases defined in the project glossary. Let $S_{PG}=\{k_1,k_2,\ldots,k_m\}$ be the set of glossary terms in the project glossary and let $S_{PH}=\{t_1,t_2,\ldots,t_n\}$ be the set of terms contained in phrases in the project glossary. The new probability score $p_{PG}(d|q)$ is defined as follows:

$$p_{PG}(d|q) \propto p(d|q)+p_f(d|q)+ \delta\left(\sum_{k_i\in S_{PG}}\frac{p(d|k_i)p(q|k_i)}{p(q)}+\sum_{t_i\in S_{PH}}\frac{p_f(d|t_i)p_f(q|t_i)}{p(q)}\right) \quad (2.6)$$

The expression above assumes that the contribution of the information from phrasing and project glossary to $p_{PG}(d|q)$ is additive. If no terms or phrases defined in the project glossary co-occur in q and d, then the probability score $p_{PG}(d|q)$ is equal to the simpler phrasing-based probability score $p_{PH}(d|q)$ defined in expression (2.4). The contribution to the overall probability $p_{PG}(d|q)$ of the project glossary keywords and phrases depends on the chosen $\delta\geq 0$. In our experiments, the parameter δ was set equal to 0.5 as it achieved the highest precision in the results. Notice that in expression (2.6) $p_{PG}(d|q)$ is defined to be proportional to the right-hand side of (2.6) which may take values larger than one. Values for $p_{PG}(d|q)$ that obey standard probability constraints (i.e. values vary in [0,1] and sum up to one for all traces to q) can be computed after a simple rescaling.

Intuitively the three enhancement methods *TC*, *Phrasing* and *Project Glossary* can be implemented synergistically by applying first the project glossary along with phrasing, and then the TC approach. Experiments that are not reported in this paper were also conducted to evaluate the performance of applying the three enhancement methods in different orders; however the combined approach described in this paper consistently outperformed the other approaches for the available datasets. The enhanced probability $P_{TCPG}(d|q)$ between two artifacts q and d that incorporates all three approaches is calculated similarly to expression (2.2) for basic term coverage, except that $p(d|q)$ is replaced by the new probability score $p_{PG}(d|q)$ defined in expression (2.6). The expression is defined as follows:

$$p_{TCPG}(d|q)=\begin{cases} m\times p_{PG}\,(d|q) & \text{if } p_{PG}(d|q) < 1/m \\ 1 & \text{if } p_{PG}(d|q)\geq 1/m \end{cases}$$

3 Evaluation

3.1 Datasets Used in the Experiments

An extensive empirical study was conducted to analyze and compare the retrieval performance of the five probabilistic retrieval methods described in Section 2: Term Coverage (TC), Phrasing, Term Coverage plus Phrasing (TCPH), Phrasing with Project Glossary (PH Glossary) and Phrasing with Project Glossary and Term Coverage (TCPH

Glossary). The experiments used five datasets containing a variety of software systems including research projects, student assignments, and industrial applications. Details of these datasets are listed below:

1. *Ice-breaker System (IBS)* describes the requirements and design of a public-works department system for managing roads de-icing. The project was initially described in "Mastering the Requirements Process" (Robertson and Robertson 1999) and then enhanced from materials found at several public works websites (Cleland-Huang et al. 2005a).
2. *Event-Based Traceability (EBT)* system, which was initially developed at the International Center for Software Engineering at the University of Illinois at Chicago (Cleland-Huang et al. 2005b), provides a dynamic traceability infrastructure based on the publish-subscribe scheme for maintaining artifacts during long-term change maintenance.
3. *Light Control System (LC)* was reconstructed from a well documented system developed at the University of Kaiserslautern (Borger and Gotzhein 2000). This system controls the lights in a building based upon user defined lighting schemes, building occupation, and current exterior illumination.
4. *SE450 Student Projects* dataset contains 15 anonymous student term projects for a MS level Software Engineering class at DePaul University. The students were given a set of requirements and instructed to implement a mid-sized traffic simulation system using the Java programming language.
5. *CM1* is a large dataset extracted from a NASA project for a science instrument which has been made available to the public on the Promise Data Repository (PROMISE 2008). This dataset was used extensively by Hayes et al in previous traceability experiments (Hayes et al. 2006).

These datasets can be considered representative of the tracing tasks in real software systems as they contain a variety of software artifact types including requirements, design documents, and source code as displayed in Table 2.

Each dataset has an associated trace matrix that explicitly defines the correct traces between the software artifacts in a system, and is used to evaluate the retrieval accuracy of the automated retrieval approach. Trace matrices for IBS, EBT, and LC were constructed and validated by researchers in our group. Each of these matrices has been used in prior work and has therefore undergone a rigorous and lengthy evaluation process. SE450 trace matrices were built by individual students in the SE450 course and then evaluated and corrected by two independent research assistants. The CMI matrix was provided by NASA and evaluated and refined by Hayes et al (2006).

Table 2 Datasets summary

Dataset	# of requirements	Document type	# of documents	# of true links	Available Project Glossary
IBS	164	UML classes	71	420	Yes
EBT	41	UML classes	52	135	No
LC	34	UML class	25	91	No
SE450	46	Java classes	475[a]	1252[a]	Yes
CM1	235	Low-level requirements	220	361	No

[a] SE450 dataset aggregated statistics from 15 student projects

3.2 Metrics

The precision of trace retrieval algorithms is generally measured at high recall levels of 80% to 90%, obtained by retrieving a relatively large set of candidate links. However, in some situations, the precision at these high recall levels is insufficient to measure local changes in the results. For example, two algorithms A and B may achieve the same overall precision at a fixed high recall level, but links generated by algorithm B may contain more correct traces near the top of the ordered list In this case, precision at lower recall levels i.e. at the top 10% or 20% of the ordered list of links, for algorithm B would be higher than that of algorithm A. Thus measuring only the overall precision would fail to capture the fact that algorithm B placed more good links higher in the ordered list than algorithm A. To capture such changes in the internal structure of the retrieved links list, the precision metric is computed at different recall levels. The precision values at 10% and 20% recall levels are especially meaningful as they measure the proportion of correct traces among the top retrieved links which represent the set of links that will be seen and inspected by the analysts first.

The *Average Precision Change (AP)* at various recall levels is a new metric which provides a more accurate measurement of the precision changes in the retrieved links list than the overall precision reported for only one specific recall level. Let Δ_i (i=1,2,…,k) be the precision change after applying two different retrieval strategies at the i-th recall level for k different recall levels. The *Average Precision Change* at various recall levels is then calculated as

$$AP = \sum_{i=1}^{k} \Delta_i / k$$

In our experiments the precision changes Δ_i are evaluated at recall levels from 10% to the highest achievable recall at intervals of 10%. Previously proposed metrics, such as DiffAR (Hayes et al. 2006) that compares the difference between the average similarity scores of correctly retrieved links and incorrectly retrieved links (false positives), are not suitable for comparing different retrieval algorithms across different projects, as these metrics take values on different scales depending on the project. In contrast, the *Average Precision Change* takes values on a fixed scale and allows the evaluation of the impact of various retrieval algorithms across different datasets.

3.3 Experimental Results

The performance of the proposed retrieval methods were compared at various recall levels for the five datasets. The retrieval algorithms employing project glossaries could only be evaluated for IBS and SE450 projects as no glossary was available for the other datasets. The discussion below focuses on the precision achieved at low recall levels (10% and 20%) corresponding to the precision of the top retrieved links that are typically presented to the analyst first, and on the precision at the highest recall levels (80% or 90%) that is an overall measure of the precision among all retrieved links.

Each graph in Fig. 1 shows the recall-precision results on a specific dataset. For convenience, the results of only one exemplary project selected among the fifteen SE450 projects are displayed in the corresponding graph in Fig. 1(b), as similar conclusions can be drawn from the remaining projects (Zou 2009).

The results show that in general the proposed enhancement approaches are effective in improving the precision of the retrieval results compared to the basic algorithm, but that the extent of the improvement differs from project to project.

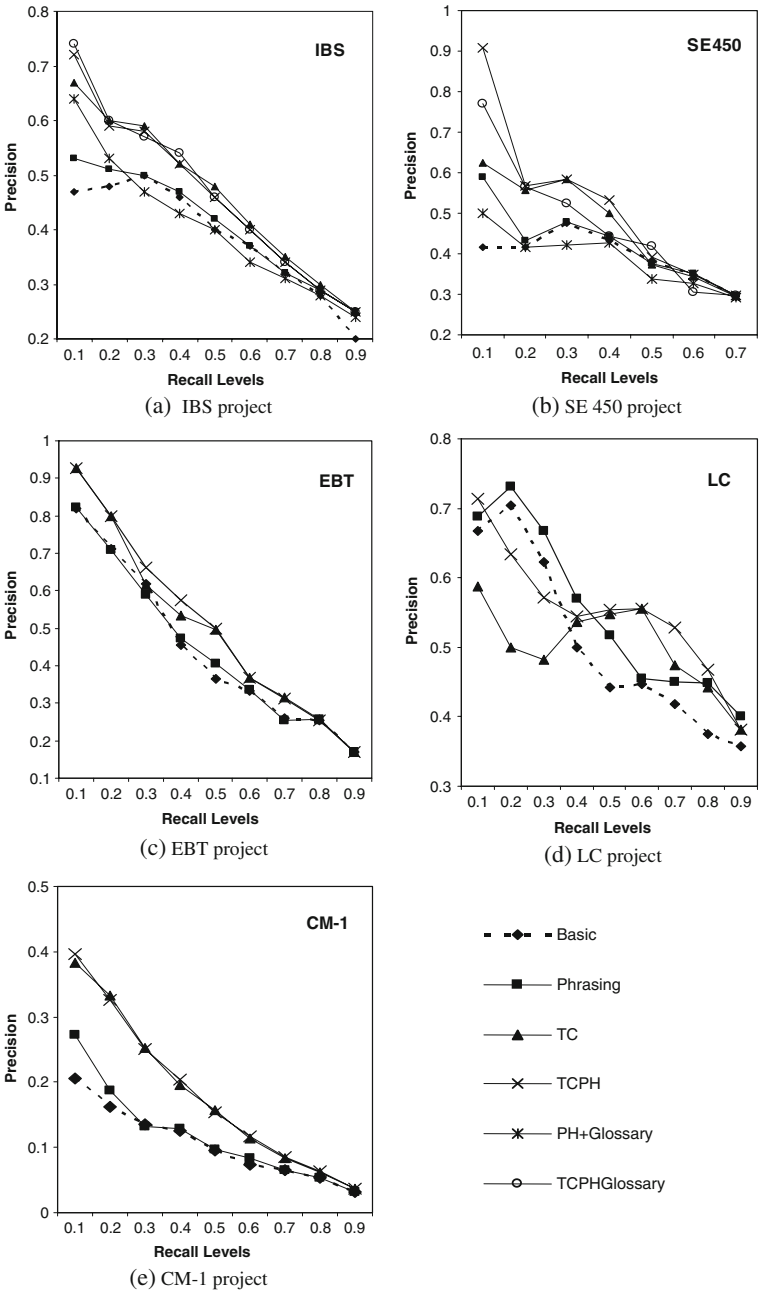


Fig. 1 Precision values for various enhancement methods applied to the IBS, EBT, LC, CM-1 and SE450 projects

Among all methods, the synergistic application of TC and phrasing together is shown to be the most effective in significantly improving precision at 10% and 20% recall levels for almost all datasets. When a meaningful project glossary is available as in the IBS dataset, the synergistic application of the TC, phrasing and project glossary methods yields the highest increase in precision among the top ranked retrieved links. The impact of the synergistic approaches that incorporate multiple enhancement methods seem to be closely related to the effectiveness of individual enhancement methods. This is evidenced in particular in the SE450 dataset, where the project glossary approach has a negative effect on the precision of the retrieval results.

The TC method shows consistent improvement in precision when applied across all datasets compared to the basic PN model. For example in IBS the precision at 10% recall increased significantly by 20% after using the TC method, and at other recall levels the increase in precision was also substantial ranging from 2% to 12% (as displayed in Fig. 1 (a)). The only exception was observed in the LC dataset (see Fig. 1(d)), where the TC method improved the overall precision at higher recall levels (from 40% to 90% recall), but lower precision was observed among the top retrieved links (10%, 20% and 30% recall). The problem for the LC dataset was caused by the occurrence of longer phrases containing terms that could be used individually to refer to different concepts. For example, the requirement "*The chosen light scene can be set by using the room control panel*." was incorrectly linked to the class "*Admin Control Panel GUI*" by the TC approach because of the three shared terms "*room*", "*control*" and "*panel*". In the requirement the three shared terms "*room control panel*" refers to a specific concept, while in the class documents, the terms "*control panel*" and "*room*" appear in different contexts and are not conceptually related. In fact when phrasing is applied with TC, this problem is partially addressed and the precision among the top retrieved links (10% recall level) increased significantly.

The phrasing algorithm achieved considerable improvement on precision for IBS, LC and CM1. The improvement resulting from phrasing is generally less significant than the TC method. In the EBT dataset, the effect of using phrasing was almost unnoticeable. Some projects in the SE450 dataset even experienced decrease in precision when phrasing was applied. Further details about the results for other SE450 datasets are reported in Section 4.1.

Similarly the effect of applying the glossary approach has not been consistent. When the glossary approach is applied, the precision at 10% and 20% recall levels increased significantly for the IBS dataset, but decreased up to 16% for the SE450 project. The analysis of the traces retrieved for the SE450 projects revealed that the project glossary approach assigned high relevance scores to links between unrelated pairs of requirements and Java classes because of the co-occurrence of weak glossary terms, such as "*vehicle*", that were inconsistently used in the two documents collections to indicate different concepts. Such incorrect traces appearing among the top retrieved links caused the decrease in precision for low recall levels.

These results have shown that the enhancement methods may be more effective for certain datasets than for other ones. This observation motivated the following research questions: "*Is there a set of characteristics in the individual projects that impacts the effectiveness of these enhancement approaches?"* In other words, "*Can we predict whether an individual approach will be effective in a given project prior to running any retrieval algorithm?"*

Section 4 examines several characteristics of software projects and proposes metrics that may be used to predict the effectiveness of the enhancement strategies for a specific project.

4 Predictors for the Enhancement Methods

Document characteristics may vary greatly from one project to another. For example, documents may be of different sizes, types, and lengths, and will inevitably use different vocabulary. Such differences can affect the performance of the various retrieval approaches. A set of metrics and dataset characteristics has been identified as possible predictors for the effectiveness of the enhancement approaches. The predictors can be used to identify which enhancement algorithm should be used in a tracing tool to improve the retrieval performance for specific document collections.

4.1 Predictor for TC and Phrasing Approaches

An intuitive metric for the term coverage approach is developed using the Query Term Coverage defined in Section 2.2. For instance, in the IBS dataset the correctly retrieved links exhibited an average higher Query Term Coverage than that of the incorrectly retrieved traces, and the TC approach achieved consistently higher precision than the basic algorithm. Thus we can deduce that the TC approach is more effective when queries have higher Query Term Coverage. This association is also seen in the LC dataset, where the TC approach performed badly. Among the top 100 retrieved links in LC, the correct traces have a lower average Query Term Coverage value than the incorrectly retrieved traces.

A project-level predictor for the effectiveness of the TC method for a given project p is computed as the *Average Query Term Coverage* of p for all artifacts q_i to be traced to artifacts d_j. The predictor denoted by *QTC(p)* is defined as follows:

$$QTC(p) = \frac{\sum_i \sum_j TC(q_i, d_j)/n_j}{n_q} \tag{4.1}$$

where $TC(q_i, d_j)$ is the Query Term Coverage value defined in expression (2.3), n_j and n_q are the total numbers of artifacts q_i and d_j in project p, respectively. Only artifacts pairs that share two or more distinct terms are considered in the above calculation, as the TC method has no impact on pairs that have only one term in common.

A metric for predicting the effectiveness of the phrasing approach in a given project is proposed in a similar fashion. The *Phrasal Term Coverage* for two artifacts q and d is defined as $PC(q, d) = \frac{m}{t_q}$ where m is the number of terms in phrases that are shared by q and d, and t_q is the total number of distinct phrasal terms in q. The Phrasal Term Coverage emphasizes the extent to which phrases are contained in linked artifacts, and takes a maximum value equal to one when all phrases in the artifact q are found in d.

The A*verage Phrasal Term Coverage* of a project p, denoted as *PTC(p)*, is defined as the average Phrasal Term Coverage $PC(q_i, d_j)$ for all artifacts q_i and d_j and is computed as follows:

$$PTC(p) = \frac{\sum_i \sum_j PC(q_i, d_j)/n_j}{n_q} \tag{4.2}$$

where n_j and n_q are defined as above. Thus if a project has high *Average Phrasal Term Coverage* value, then the phrasing algorithm is expected to retrieve a higher proportion of correct links and therefore increase the precision of the retrieved traceability links.

4.2 Evaluating the Predictors for the TC and Phrasing Approaches

The *Average Query Term Coverage* and the *Average Phrasal Term Coverage* metrics defined in Section 4.1 were computed for each of the available datasets: IBS, EBT, LC, CM-1 and the fifteen SE450 projects. The performance of both the TC and the phrasing methods for each project were compared against the basic PN algorithm by measuring the average precision change at various recall levels achieved by the enhancement retrieval methods.

The association between the *QTC* and improvement in the precision is depicted in Fig. 2(a), where the points correspond to the nineteen projects used in the analysis. The x-axis represents the *QTC* values while the y-axis represents the average precision change achieved by the TC approach. The scatterplot in Fig. 2(a) shows a strong positive association between the two variables, indicating that higher average precision changes are typically associated with higher *Average Query Term Coverage* values. Similarly the scatterplot in Fig. 2(b) displays a positive association between the *Average Phrasal Term Coverage* (*PTC*) metric and the average precision change achieved by the phrasing approach. A simple regression analysis also shows that the pair-wise associations between the predictors' values and the average precision change values are significant at 5% level.

The patterns displayed in the graphs in Fig. 2. indicate that both the TC and phrasing enhancement methods are more likely to effectively increase the precision of the retrieval results in projects that exhibit higher QTC or PTC metric values, especially for QTC values higher than 0.3 and PTC values higher than 0.2.

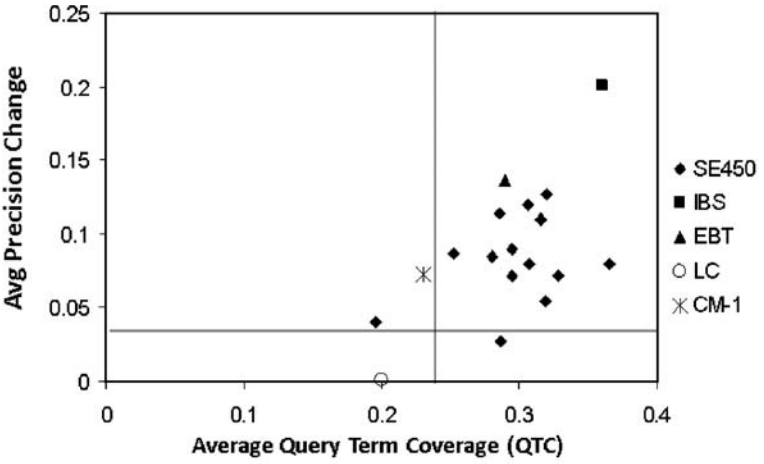
A simple illustration of how the two metrics may be used to predict the effectiveness of the enhancement approaches is described in a case study using the nineteen available projects. The heuristic approach defines thresholds for the QTC and PTC metrics that are used to determine if a given enhancement method is likely to be effective on a given project.

The thresholding approach for QTC and PTC metrics follows a simplified version of an algorithm proposed by Cronen-Townsend et al. (2002) to determine the threshold for a clarity score metric measuring the performance of queries in general IR searches. In our case study, thresholds are determined using the following two-step procedure:

Step 1: Select the top 90% of projects in the training set ranked according to the average precision change achieved with the enhancement method (TC or phrasing approach). This step rejects the bottom 10% of the projects for which no significant precision change was observed.

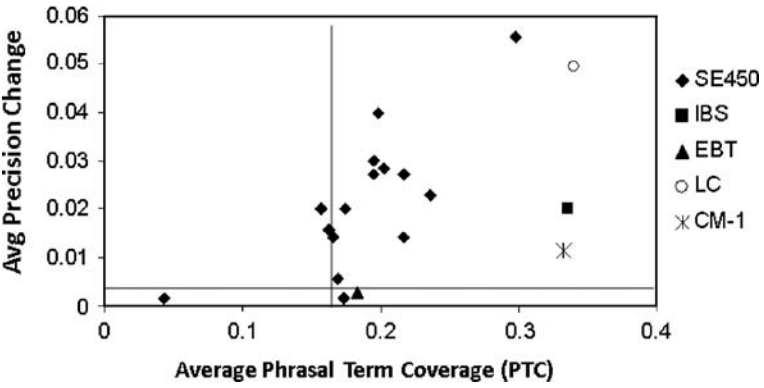
Step 2: Compute the QTC and PTC predictor values for all projects selected in Step 1 and set the metric threshold for each metric equal to the top 80% of the computed predictor values.

Figure 2 shows the application of the thresholding method to the 19 projects. The horizontal lines in the graphs in Fig. 2a and b, delimit the subsets created in step 1, and the vertical lines represent the thresholds computed in step 2. Based on the heuristic classification method, the TC method is expected to be effective for any project with *QTC* value of at least 0.24, while the phrasing approach is expected to improve the precision of retrieved links if a project has a *PTC* value of at least 0.17. The percentiles used in the two steps of the thresholding procedure were determined empirically. The percentage values can be set at higher levels if more conservative threshold values are needed. The heuristic approach was validated using a leave-one-out cross validation technique that computed the threshold for Phrasing using 18 projects in the training set



Correlation value= 0.583

(a) QTC values vs precision change for TC approach



Correlation value= 0.537

(b) PTC values vs precision change for phrasing approach

Fig. 2 Association between predictors and average precision change

and applied it against the remaining project. The process was repeated 19 times and each project was used once for testing. The results are shown in Table 3. As there is only one available project for which the TC approach did not yield any significant improvement in precision, the classifier could not be tested on the TC approach. The heuristic procedure for computing the PTC threshold appears to be helpful, as the classifier correctly identified 80% of the projects for which the phrasing method is effective and 50% of the projects in which phrasing was not helpful.

The thresholds obtained in this study can be used to predict if the enhancement methods can be effective in improving the accuracy of the retrieval results. In Section 4.3 below, we propose an iterative approach that uses the threshold values defined here to initially select an enhancement tracing method, and then incorporates user's feedback to improve and refine the choice of the tracing tool to use in automated trace retrieval.

Table 3 Leave-one-out cross-validation results using PTC values to predict Phrasing performance in the 19 available projects

	Projects Predicted as			Total
		Effective	Not-Effective	
Actual	Effective	12	3	15
	Not-Effective	2	2	4
Recall=0.80% Precision=0.86				

4.3 An Iterative Approach of Applying the Predictors

In practice software artifacts and associated traces are often built incrementally. This section therefore describes an iterative technique that determines which trace retrieval approach is more effective in a new project when no prior traceability knowledge is available. In these circumstances a partial answer set is constructed in real-time from the user's feedback, and the enhancement strategies are then turned on or off according to this feedback. Furthermore the iterative approach enables an enhancement strategy to be turned on or off as more information becomes available and it becomes clearer whether a particular strategy improves or detracts from the quality of the trace results.

The iterative technique outlined in Fig. 3 starts by utilizing the predictor values to select the automated retrieval algorithm to use for a new project. For instance if both the TC and the phrasing predictors exhibit high values, then the synergistic approach incorporating both enhancement methods will be applied to retrieve traces. The selected algorithm is then used to compute the link probability scores for the project artifacts, and the top K links ranked according to their probability scores are presented to the user for evaluation. The user accepts or rejects the presented links, and the information is stored in an answer set.

In the next step the answer set is used to determine if the selected enhancement method is appropriate. Probability scores are computed for each of the links in the partial answer set using the basic PN model scores as well as each of the enhancement techniques. An enhancement method is considered effective and will be selected for the subsequent tracing task, if it increases the average probability scores for true links more than for false links.

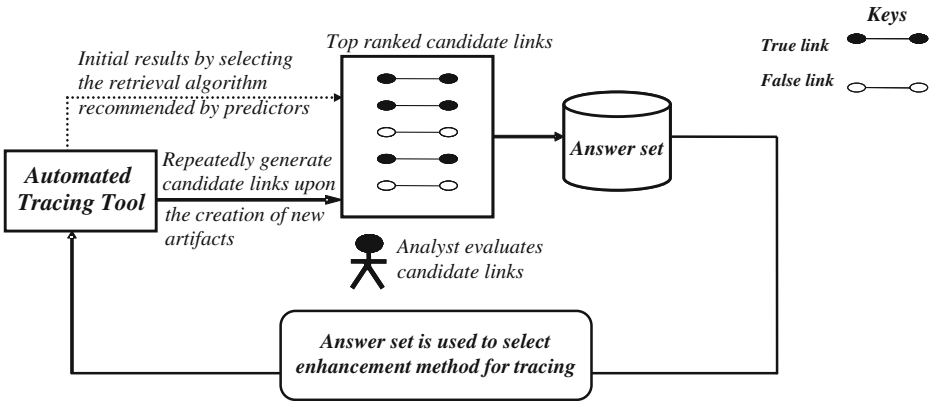


Fig. 3 The iterative approach of applying predictors in a given project

This metric was preferred to the standard recall and precision metrics because it more accurately evaluates the performance of retrieval methods for small sets of traces.

Notice that the retrieval tool is expected to be used repeatedly to generate potential traces whenever new requirements or new artifacts are created during the project development life cycle. Thus the answer set is iteratively augmented by adding new user's evaluations prior to each run.

The application of this approach is illustrated for the SE450 fifteen projects. At first 10% of the requirements were randomly selected for each project to simulate the initial state of the iterative user feedback algorithm, and then 5% of requirements were added at each later iteration. The choice of 5% was primarily for convenience as it allowed us to evaluate the results of the iterative approach at a finer scale. User's feedback was collected for $K=20$ or $K=50$ top ranked candidate links representing links between the selected requirements and all traceable documents. User feedback was simulated using the original traceability matrices supplied with the 15 projects.

Two metrics were used to evaluate the accuracy of the prediction based on the answer set: True Positive (TP) rate that represents the proportion of projects for which the enhancement methods are correctly predicted to be effective, and False Positive (FP) rate that represents the proportion of projects for which the enhancement methods are incorrectly predicted to be effective. The results are displayed in Table 4. In both experiments for $K=20$ and $K=50$, when the initial answer set is small, the TP rate is about 50% indicating that for only half of the projects the approach is able to predict correctly the effectiveness of either the TC or phrasing enhancement methods. As the answer set is augmented with additional user's feedback, the FP rate decreases while the TP rate increases, indicating, as expected, that a larger answer set is able to provide more accurate predictions about the enhancement methods performance. This suggests that enhancement strategies should only be activated once a sufficient body of data has been collected. As expected the iterative approach is consistently more accurate for $K=50$.

5 Evaluating the Effectiveness of Project Glossaries for Trace Retrieval

The experiments discussed in Section 3.3 show that the use of project glossary information does not necessarily improve the probabilistic retrieval tool accuracy, and that the results are dependent upon the quality of the project glossary. Zou et al. (2008) identified three characteristics of a project glossary that could be used to predict when the glossary approach can be effective in improving the trace precision for a specific project. These characteristics are discussed below.

Table 4 Results from the iterative approach in SE450 projects

	User Feedback for Top 20 links								
Candidate links list size	10%	15%	20%	25%	30%	35%	40%	45%	50%
False Positive (FP) rate	0. 51	0.33	0.44	0.33	0.33	0.19	0	0	0
True Positive (TP) rate	0.51	0.82	0.74	0.85	0.74	0.92	0.89	0.93	1
	User Feedback for Top 50 links								
Candidate links list size	10%	15%	20%	25%	30%	35%	40%	45%	50%
False Positive (FP) rate	0.62	0.39	0.39	0.35	0.33	0.33	0.33	0	0
True Positive (TP) rate	0.58	0.90	0.93	1	1	1	1	1	1

5.1 Criteria to Evaluate Project Glossaries for Trace Retrieval

Criterion # 1: Project glossary items should be consistently used in the traced documents A project glossary describes the terminology used in a specific software project, and is intended to facilitate the consistent use of terms across all project artifacts. However in practice, project glossaries are often not used consistently and requirements specifications and other software documents contain synonyms of the glossary terms. When this happens, project glossaries may have insignificant impact, or even no impact upon the quality of the retrieval results. Therefore, the presence of synonyms of glossary items in the traced documents provides a strong indication that the glossary may not have been used consistently. A simple method to detect synonyms of glossary terms is implemented using WordNet (Fellbaum 1998) which is a semantic dictionary in which words are organized into logical groups consisting of related synonyms.

Criterion # 2: Glossary items should have high term specificity Term specificity indicates the quality of a term in describing the document content, and is commonly computed using *idf* (*inverse document frequency* (Joho and Sanderson 2007). Thus glossary terms specificity is measured as $idf(t) = ln(|R|/|R_t|)$, where $|R|$ is the total number of requirements and $|R_t|$ is the number of requirements containing t. Glossary terms with high specificity values occur in fewer requirements, and are more useful for identifying a specific concept, and hence for retrieving documents related to that concept.

Criterion # 3: Glossary items should be domain specific Domain-specific terms occur more frequently in project specific documents, and are often associated to critical concepts of the project. The domain specificity *DS(t)* for a term t is computed as follows:

$$DS(t) = \ln\left(\frac{freq(t,R)}{\sum_{t\in D} freq(t,R)} \Big/ \frac{freq(t,G)}{\sum_{t\in G} freq(t,G)}\right) \tag{5.1}$$

where *freq(t,R)* is the frequency of term t in the requirements collection R associated with the project glossary, and *freq(t,G)* is the frequency of term t in the general technical corpus G that contains requirements from various domains. In our experiments the corpus contains thirty eight sets of Software Requirement Specifications (SRS) taken from a variety of software projects. In addition to the five datasets that are introduced in Section 3.1, the corpus also includes projects ranging from industrial applications to research projects. Project topics include NASA's Moderate Resolution Imaging Spectrometer, an industrial production lines construction system, vehicle parts finder, meetings scheduler, battleships game, and an enterprise level service bus scheduling system. If a term is unique to the specific project, i.e. *freq(t,G)= 0, DS(t)* is assigned a large value.

5.1.1 Applying the Criteria on Project Glossaries

The three criteria are applied to evaluate the impact of using a project glossary to increase the precision of the retrieval results. Thus the project glossary can be considered weak with respect to its ability to improve the retrieval results for the project if synonyms of glossary terms are used, or if glossary terms have an average low specificity and low domain-specificity.

The IBS project and the SE450 fifteen projects were the only projects supplied with a project glossary. The IBS glossary has six keywords and 28 phrases, while the SE450 glossary contained four keywords and six phrases. On the basis of the proposed criteria, the SE450 project glossary was found to be inconsistently used in the fifteen projects, while the IBS project glossary was found to be more meaningful. Synonyms of glossary terms were frequently used in the SE450 Java classes, for instance '*car*' and '*obstruction*' were used in place of '*vehicle*' and '*obstacle*'. No synonyms of glossary items were detected in the IBS dataset. Both the average term specificity and the average domain specificity of the SE450 glossary items were significantly lower than the corresponding values for IBS glossary terms (Zou et al. 2008). The average term specificity and the average domain-specificity of terms in the SE450 project glossary were 0.78 and 5.03, respectively. Both scores are about 50% lower than the corresponding scores of 1.63 and 7.69 for the IBS glossary.

The weakness of the SE450 projects glossary is confirmed by the results in Section 3.3 for the SE450 datasets which show that the glossary approach reduced the accuracy of the tracing tool retrieval results. On the contrary for the IBS dataset, the glossary approach was able to retrieve a larger proportion of correct traces. These results suggest that the three proposed criteria provide a simple and effective way to predict when a project glossary can be used to improve the accuracy of the retrieval tool.

5.2 A Method for Automatically Extracting Keywords and Phrases

This section presents an automated technique for extracting a set of important keywords and phrases from the project requirement specifications that can be used in lieu of a project glossary to help improve the precision of the retrieval algorithm. This technique was originally designed to be used when glossaries are either not supplied with the software projects or are inconsistently followed during the development phase. Several methods for extracting keywords from document collections have been proposed in IR. Some techniques are based on statistical approaches that identify significant terms on the basis of term frequency (Matsuo and Ishisuka 2004), but ignore the syntactical meaning of the terms. In automated tracing, a technique for keyword extraction based on Matsuo and Ishisuka's approach was already applied to rank terms in artifacts according to their perceived importance for trace recovery (Dekhtyar et al. 2007). Our proposed extraction method applies a syntactical method to identify critical keywords and phrases from the requirement specifications. The syntactical approach enables the tool to extract only single nouns and two-noun phrases which were found in prior work to be the most common type of phrases found in a project glossary. The approach consists of the following two steps:

Step 1. *Generate candidate keywords and phrases*. Candidate items including single nouns and two-noun phrases are identified by applying a POS tagger such as Qtag to the set of requirements specifications.

Step 2. *Filtering*. Filters are applied to remove unimportant items from the list generated in Step 1.

The following three filters are applied:

Filters A and B: Term and Domain specificity. Keywords with term and domain specificity values below certain thresholds will be removed, as they might decrease the precision of the trace retrieval results. The threshold values are set by the analyst and depend on document collection characteristics.

Filter C: Noun filtering. A single noun that is included in a candidate phrase as head noun is removed because the phrase is considered more meaningful at representing project specific concepts. For example, '*truck'* in the phrase "*truck list*" is used to modify the head noun "*list*". It is necessary to remove "*list*" from the candidate terms set as "*truck list*" is considered more specific than single noun "*list*",

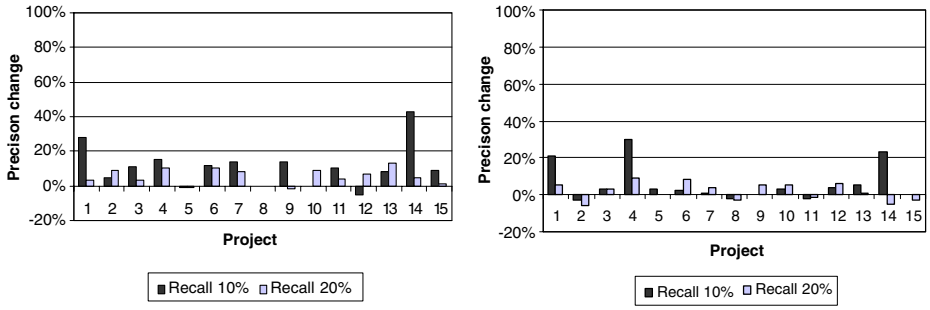
The resulting list will consist of single nouns and two-noun phrases that have high term and domain specificity.

5.2.1 Evaluating the Extraction Method

The automated extraction method was applied to the requirements collections for the SE450 dataset and the EBT, LC and CM1 datasets. In our experiments, Filter (A) in step 2 removed a low term specificity keyword if it occurred in at least three requirements for projects containing less than 60 requirements, or in at least 5% of the total requirements for larger projects. The filtering step (B) based on domain specificity retrieved 1) all unique items for the project (i.e. $freq(t,G)=0$), and 2) the top 50% items with the highest domain-specificity score. The thresholds in filters A and B were selected on the basis of an exploratory study that analyzed the distribution of glossary terms in the requirements documents for the available projects. Most of the items in the IBS project glossary occurred in no more than three requirements, while many items of the project glossaries associated with the SE450 dataset occurred in a much larger number of requirements. Preliminary results suggested that the selected threshold values are appropriate for extracting critical phrases and keywords that can be used effectively to improve the accuracy of the retrieval results.

The extracted set for the SE450 dataset contained five keywords and 30 phrases, but had only two terms in common with the existing project glossary. For EBT and LC, the set contained eight keywords and 15 phrases, and six keywords and 11 phrases respectively. For the large-scaled CM1 dataset, the method extracted 59 keywords and 164 phrases.

The project glossary approach was then applied to the datasets using the extracted critical set. The two graphs in Fig. 4 display changes in precision at 10% and 20% recall for the fifteen SE450 projects comparing the Project Glossary approach incorporating the extracted keywords set with (a) the basic PN tracing technique, and with (b) the Project Glossary algorithm using the existing project glossaries. Both graphs show that the extracted keywords set was, in most cases, more effective in improving the precision of the



(a) compared to basic PN **(b) compared to the existing glossary usage**

Fig. 4 Precision change at 10%, 20% recall for algorithm using extracted set on SE450

tracing results, even when compared with the existing project glossary. Similarly, results for the EBT, and CM1 datasets displayed in Figs. 5 and 6 show that the extracted keywords set was effective in improving the retrieval accuracy, especially among the top retrieved links. Figure 7 shows that for the LC dataset the keyword extraction method achieves the same or slightly better accuracy than the phrasing algorithm at most recall levels.

6 Threats to Validity

In this section we discuss reservations and threats to validity that can affect the generalization of our results.

6.1 Accuracy in Tracing Results

The validity of our conclusions and the results of our experiments are affected by the correctness of the answer sets used to evaluate and compare the accuracy of the automated tracing tools. Each dataset used in our experiments was provided with a trace matrix that defines the set of traceability links between the artifacts in the system. As described in Section 3, the trace matrices were built by various software engineers that manually traced the artifacts in each dataset. There is always a certain level of subjectivity in the evaluation of traces, and it is possible that analysts may have missed some traces, or have incorrectly included irrelevant traces. To mitigate the risk of errors in the answer sets, the trace matrices of all datasets were refined and re-evaluated thoroughly by several researchers while conducting the experiments. However, it is still possible that the answer sets are either incomplete or contain incorrect traces. Thus the recall and precision values in our experiments should be more precisely interpreted in terms of the traces identified as "correct" through manual tracing.

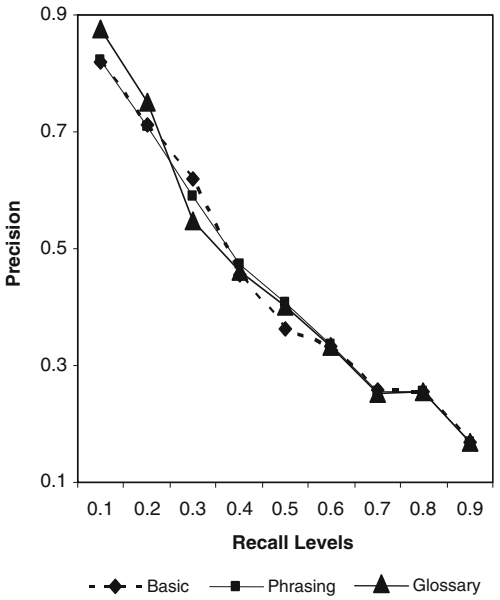


Fig. 5 Effect of extracted set on EBT

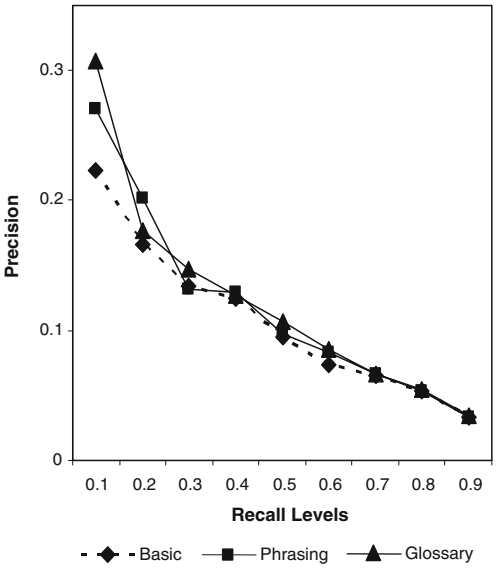


Fig. 6 Effect of extracted set on CM-1

6.2 Available Datasets for Experiments

One important constraint on our experiments is caused by the limited number of available projects that can be used to test our approaches. Because of the lengthy process to build accurate trace matrices, it is difficult to create datasets for traceability experiments. Our

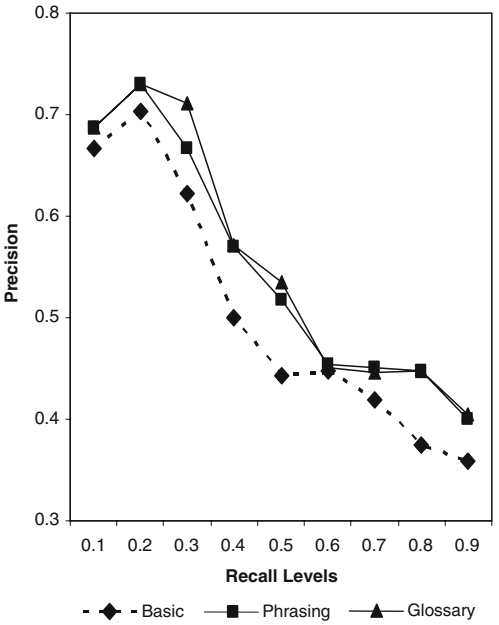


Fig. 7 Effect of extracted set on LC

experiments focus only on traceability from requirements to Java and UML classes and from higher level requirements to lower level requirements. Since we did not have additional datasets with associated traceability matrices, we could not evaluate the effectiveness of our approaches for traces between other artifacts. It would be interesting to evaluate if the impact of the three enhancement methods and the artifact textual characteristics measured by Query Term Coverage and Query Phrasal Coverage vary by artifact type.

6.3 Use of Predictors

Another validity concern deals with the two heuristic approaches, described in Section 4, that use the QTC and PTC predictors to select the enhancement methods that are more effective in improving the retrieval results for a given project.

The first heuristic approach computes threshold values for the QTC and PTC metrics assuming that projects with known traces are available. This may have limited practical value, since traces are often not known a priori. The threshold values identified in our experiments have not been tested on other projects. The leave-one-out cross-validation procedure is an attempt to check the validity of the threshold value for the PTC metric. The results of this validation procedure might show some bias since fifteen of the 19 projects have strong commonalities. They are fifteen students projects produced as part of the same course, and with the same set of requirements.

An additional concern regards the results for the iterative procedure proposed in Section 4 to select the effective enhancement procedures. Users' feedback is simulated using the known answer set, and therefore it does not consider that the software engineer may miss some links, or trace incorrect artifacts during the manual tracing process. The results from the iterative approach described in Table 4 could therefore overestimate the accuracy of the iterative procedure at the various steps, and less accurate feedback could result in more frequent switches between enhancement strategies.

7 Conclusions

This paper has discussed and compared three enhancement strategies that can be incorporated either individually or synergistically into the basic PN model in order to improve the precision of the trace retrieval results. These methods are easy to implement and require no extra human effort. Although the results have shown that the performance of these retrieval algorithms varies from project to project, they have proved the general effectiveness of using such enhancement methods to improve retrieval results especially among the top retrieved links. Enhancement methods are less effective in increasing the precision at high recall levels, since the low-ranked missed traces often represent requirements and documents that share very few or no terms and phrases. In these cases term-based retrieval approaches are ineffective.

The paper has also provided an automated method for extracting keywords and phrases from existing requirements collections in a project with weak or missing glossaries. The retrieval algorithm using either the project glossary information or the set of extracted key terms and phrases generally improves precision among the top ranked set of retrieved links. Although the proposed enhancement methods can increase the analyst's trust in the tracing tool, some true links are still missed, and are hard to retrieve using only textual content information.

With the assistance of the proposed prediction models that utilize the predictors for individual enhancement methods, an automated tracing tool can make real-time decisions on whether to apply a certain method in order to improve results. Results of a preliminary study indicate that the predictor values can provide useful guidelines to select a specific tracing approach when there is no prior knowledge on the 'answer set' for a given project. However more extensive experiments are necessary to test and validate the predictor models on a larger number of datasets.

The effectiveness of the predictor models can be improved by evaluating users' feedback to determine when to switch an enhancement model on or off. This feedback is gathered from the users as they issue and review traces. As a result the trace matrix will be gradually generated as a natural byproduct of the users' use of the trace tool. Furthermore, the benefits of improving the precision in a given project will be experienced throughout the remaining lifetime of the software system, and could significantly alleviate future maintenance efforts.

Acknowledgments The work described in this paper was partially funded by NSF grants CCR-0306303 and CCF0810924.

References

Antoniol G, Canfora G, De Lucia A, Casazza G (2000) Information Retrieval Models for Recovering Traceability Links between Code and Documentation. Proceedings of the International Conference on Software Maintenance, San Jose, California, USA, pp. 40–51.

Borger E, Gotzhein R (2000) Requirements Engineering Case Study 'Light Control'. Journal of Universal Computer Science 6(7):580–596

Burke R, Hammond K., Kulyukin V., Lytinen S., Tomuro N. and Schoenberg S. (1997) Natural language processing in the FAQ finder system: results and prospects. AAAI Spring Symposium on Natural Language Processing for the World Wide Web, pp. 17–26.

Church K, Hanks P (1990) Word Association Norms, Mutual Information, and Lexicography. Computational Linguistics 16(1):22–29

Cleland-Huang J, Settimi R, BenKhadra O, Berezhanskaya E, Christina S (2005a) Goal-Centric traceability for managing non-functional requirements. Proceedings of the 27th International Conference on Software Engineering, St. Louis, MO, USA, pp. 362–271.

Cleland-Huang J, Settimi R, Duan C, Zou X (2005b) Utilizing supporting evidence to improve dynamic requirements traceability. Proceedings of the 13th IEEE International Requirements Engineering Conference, Paris, France, pp. 135–144.

Croft W, Turtle H, Lewis A (1991) The use of phrases and structured queries in information retrieval. Proceeding of the 14th International ACM SIGIR conference on Research and development in information retrieval, Chicago, IL, USA, pp. 32–45.

Cronen-Townsend S, Zhou Y, Croft W B (2002) Predicting Query Performance. Proceedings of the 25th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2002), pp 299–306.

Davis AM (1990) Software Requirements: Analysis and Specification. Prentice Hall, Englewood Cliffs, NJ

De Lucia A, Fasano F, Oliveto R, Tortora G (2007) Recovering traceability links in software artifact management systems using information retrieval methods. ACM Transactions on Software Engineering and Methodology (TOSEM), 16(4), article n.13.

De Lucia A, Oliveto R, Tortora G (2009) Assessing IR-based traceability recovery tools through controlled experiments. Empirical Software Engineering 14(1):57–92

Deerwester S, Dumais S, Furnas G, Landauer T, Harshman R (1990) Indexing by latent semantic analysis. Journal of the American Society for Information Science 41:391–407

Dekhtyar, A.; Hayes, J.H.; Sundaram, S.; Holbrook, A.; Dekhtyar, O., (2007) Technique Integration for Requirements Assessment, Proceedings of 15th International Requirements Engineering Conference, pp.141–150.

Evans MW (1989) The Software Factory. John Wiley and Sons, Hoboken, NJ

Fagan J (1987) Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods (Doctoral dissertation, Cornell University, Computer Science Department). Technical Report, pp. 87–868.

Fellbaum, C editor (1998). Wordnet: An Electronic Lexical Database, MIT Press Books.

Forsythe GE, Malcolm MA, Moler CB (1977) Computer Methods for Mathematical Computations (Chapter 9: Least squares and the singular value decomposition). Prentice Hall, Englewood Cliffs, NJ

Frakes WB, Baeza-Yates R (1992) Information Retrieval: Data Structures and Algorithms. Prentice Hall, Englewood Cliffs, NJ

Furnas G W, Deerwester S, Dumais S T, Landauer T K, Harshman R A, Streeter V, Lochbaum K E (1988), Information retrieval using a singular value decomposition model of latent semantic structure. Proceedings of SIGIR, pp. 465–480.

Gay L, Croft W (1990) Interpreting Nominal Compounds for Information Retrieval. Inf Process Manage 26 (1):21–38

Gotel O, Finkelstein A (1994) An analysis of the requirements traceability problem. Proceedings of the 1st International Conference on Requirements Engineering, Colorado Springs, Colorado, USA, pp. 94–101.

Hayes , J. H., Dekhtyar, A., Osbourne, J. (2003). Improving requirements tracing via information retrieval. Proceedings of the 11th International Conference on Requirements Engineering, pp. 151–161.

Hayes JH, Dekhtyar A, Sundaram S (2006) Advancing Candidate Link Generation for Requirements Tracing: the Study of Methods. IEEE Transactions on Software Engineering 32(1):4–19

Interactive Development Environments (1991). Software through pictures: products and services overview, IDE Inc.

Joho H, Sanderson M (2007) Document Frequency and Term Specificity. Proceeding of the 8th Recherche d'Information Assistée par Ordinateur Conference (RIAO'07), Pittsburgh, PA, USA.

Jones KS, van Rijsbergen CJ (1976) Information Retrieval Test Collections. Journal of Documentation 32:59–75

Kaindl H (1993) The Missing Link in Requirements Engineering. ACM SIGSOFT Software Engineering Notes 18(2):30–39

Lin J, Lin C C, Cleland-Huang J, Settimi R, Amaya J, Bedford G, Berenbach B, Khadra O B, Duan C Zou X. (2006). Poirot: a distributed tool supporting enterprise-wide traceability. Proceeding of the 14th IEEE International Conference on Requirements Engineering, Minneapolis, MN, USA, pp. 11–15.

Maletic J I, Munson E V, Marcus A, Nguyen T N (2003) Using a hypertext model for traceability link conformance analysis. Proceeding of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering, Montreal, CA, USA, pp. 47–54.

Marcus A, Maletic J I (2003) Recovering documentation-to-source-code traceability links using latent semantic indexing. Proceeding of the 25th IEEE International Conference on Software Engineering, Portland, Oregon, USA, pp. 125–135.

Matsuo Y, Ishisuka M (2004) Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. International Journal on Artificial Intelligence Tools 13(1):157–169

PROMISE (2008) Software Engineering Repository, http://promise.site.uottawa.ca/SERepository, accessed 8/8/2008.

Robertson S, Robertson J (1999) Mastering the Requirements Process, Reading. Addison-Wesley, MA

Rocchio J (1971) The SMART Retrieval System: Experiments in Automatic Document Processing (Relevance feedback in information retrieval). Prentice-Hall, Englewood Cliffs, NJ

Salton G, Buckley C (1988) Term weighting approaches in automatic retrieval. Information Processing and Management 24(5):513-523.

Salton G, Yang C, Yu C (1974) A Theory of Term Importance in Automatic Text Analysis. Journal of the American Society for Information Science 26(1):33–44

Salton G, Wong A, Yang CS (1975) A Vector Space Model for Automatic Indexing. Commun ACM 18 (11):613–620

Settimi R, Cleland-Huang J, BenKhadra O, Mody J, Lukasik W, DePalma C (2004) Supporting change in evolving software systems through dynamic traces to UML. Proceeding of the 7th IEEE International Workshop on Principles of Software Evolution, Kyoto, Japan, pp. 49–54.

Singhal A, Choi J, Hindle D, Lewis DD, Pereira F (1999) AT&T at TREC-7. Proceedings of TREC-7, Gaithersburg, MD, USA, pp. 239–252.

Tufis D, Mason O (1998) Tagging Romanian texts: a case study for QTAG, a language independent probabilistic tagger. Proceedings of the International Conference on Language Resources & Evaluation, Granada, Spain, pp 589–596

Wong SKM, Yao YY (1991) A Probabilistic Inference Model for Information Retrieval. Information Systems 16(3):301–321

Zou X (2009) Improving Automated Requirements Trace Retrieval Through Term-Based Enhancement Strategies. PhD thesis, School of Computing, DePaul University, Chicago, IL. Technical Report n. 09–001.

Zou X, Settimi R, Cleland-Huang J (2006) Phrasing in Dynamic Requirements Trace Retrieval, Proceedings of the 30th Annual International Computer Software and Application Conference (COMPSAC06). Chicago, IL, USA, pp 265–272

Zou X, Settimi R, Cleland-Huang J (2007) Term-based Enhancement Factors in Automated Requirements Traceability Retrieval, Proceedings of the 2nd International Symposium on Grand Challenge in Traceability. Lexington, KY, USA, pp 40–45

Zou X, Settimi R, Cleland-Huang J (2008) Evaluating the Use of Project Glossaries in Automated Trace Retrieval. Proceedings of the 2008 International Conference on Software Engineering Research and Practice (SERP'08), Las Vegas, USA, pp. 157–163.

Xuchang Zou received a PhD in Computer Science from the School of Computing at DePaul University in Feb. 2009. She worked as a research assistant at the System and Requirements Engineering Center of DePaul University until 2008. Her research focus is primarily on requirements engineering and she is particularly interested in the application of information retrieval and machine learning techniques in requirements engineering. Prior to coming to DePaul University, she attended XiDian University in China and received an MS degree in Computer Science. She is currently working at Microsoft on enterprise email archiving and E-discovery software.

Raffaella Settimi is Associate Professor at the School of Computing of DePaul University. She received a M.Sc. in Statistical Sciences from the University of Sheffield (UK) in 1992 and a Ph.D. in Statistics from the University of Perugia (Italy) in 1995. Her research interests include information retrieval methods, data mining techniques, Bayesian learning from large datasets and latent variable modeling. Her work on these topics has appeared in several international journals, and conference proceedings.

Jane Cleland-Huang is Associate Professor at DePaul University's School of Computing. She received a Ph.D. in Computer Science from the University of Illinois at Chicago. Her research interests include requirements traceability with emphasis on tracing non-functional requirements (NFRs) across the system lifecycle. She is a member of the IEEE Computer Society and the Software Engineering Research Consortium (SERC), and serves as regional director of the Center of Excellence for Software Traceability. Her work is currently funded by grants from the National Science Foundation and Siemens Corporate Research. Contact her at jhuang@cs.depaul.edu