On Integrating Orthogonal Information Retrieval Methods to Improve Traceability Recovery

Malcom Gethers*, Rocco Oliveto†, Denys Poshyvanyk* and Andrea De Lucia‡
*Computer Science Department, The College of William and Mary, Williamsburg, VA, USA
†STAT Department, University of Molise, Pesche (IS), Italy
‡School of Science, University of Salerno, Fisciano (SA), Italy
mgethers@cs.wm.edu, rocco.oliveto@unimol.it, denys@cs.wm.edu, adelucia@unisa.it

***Abstract*—Different Information Retrieval (IR) methods have been proposed to recover traceability links among software artifacts. Until now there is no single method that sensibly outperforms the others, however, it has been empirically shown that some methods recover different, yet complementary traceability links. In this paper, we exploit this empirical finding and propose an integrated approach to combine orthogonal IR techniques, which have been statistically shown to produce dissimilar results. Our approach combines the following IR-based methods: Vector Space Model (VSM), probabilistic Jensen and Shannon (JS) model, and Relational Topic Modeling (RTM), which has not been used in the context of traceability link recovery before. The empirical case study conducted on six software systems indicates that the integrated method outperforms stand-alone IR methods as well as any other combination of non-orthogonal methods with a statistically significant margin.**

I. Introduction

Traceability links between software artifacts represent an important source of information, if available, for different stakeholders and provides important insights during software development [1]. Unfortunately, establishing and maintaining traceability links between software artifacts is an error prone and person-power intensive task [2]. Consequently, despite the advantages that can be gained, effective traceability is rarely established.

Extensive effort in the software engineering community has been brought forth to improve the explicit connection of software artifacts. Promising results have been achieved using Information Retrieval (IR) techniques [3], [4] to recover links between different types of artifacts (see e.g., [1], [5]). IR-based methods propose a list of candidate traceability links on the basis of the textual similarity between the text contained in the software artifacts. The conjecture is that two artifacts having high textual similarity share similar concepts, thus they are good candidates to be traced on each other. Several IR methods have been employed for traceability recovery, such as Vector Space Model (VSM) [3] and Latent Semantic Indexing (LSI) [4].

The experiments conducted to evaluate the accuracy of all these methods highlight that there is no clear technique able to sensibly outperform the others. In a recent study [6] it has been empirically proved that widely used IR-based methods, such as VSM and LSI, are nearly equivalent, while Latent Dirichlet Allocation (LDA) [7]—a topic modeling technique recently used for traceability link recovery [8]—is able to capture some important information missed by the other exploited IR methods, while its accuracy is lower than that of the other IR methods.

This recent empirical result motivates our work. In particular, orthogonality of IR-based techniques may present the opportunity to improve accuracy through combining different techniques. In addition, topic modeling techniques should be further analyzed since they seem to capture a dimension missed by canonical IR methods. Thus, in this paper we propose (i) a novel method for traceability link recovery that exploits Relational Topic Model (RTM) [9] for extracting and analyzing topics and relationships among them from software artifacts; and (ii) an approach to efficiently combine different IR methods for traceability recovery. The results of the case study conducted on six software repositories indicate the benefits achieved while combining RTM with canonical IR techniques, in particular a technique based on VSM [3] and a technique based on probabilistic model, namely Jensen and Shannon (JS) [10]. The combination is highly valuable only when canonical methods are combined with the topic modeling technique based on RTM. This is because RTM is orthogonal to VSM and JS, while the latter two canonical methods provide similar results, thus confirming the finding achieved in [6]. In the context of our case study, we also analyzed the impact on the recovery accuracy of the natural language (i.e., English versus Italian) and the type of software artifacts (i.e., use cases, UML diagrams, and test cases) to be traced on source code classes. The data used in the evaluation is made freely available online, encouraging other researchers to replicate this work[1].

Summarizing, the specific contributions of the paper are:

- the definition of a novel traceability recovery method based on RTM;
- an hybrid approach for traceability recovery that combines different IR methods. The combination of orthogonal techniques provides a tangible improvement in recovery accuracy;
- an analysis on how the language and the type of the software artifacts to be traced interact with the IR method and influence the recovery accuracy

[1] http://www.cs.wm.edu/semeru/data/icsm2011-traceability-rtm

Structure of the paper. Section II presents background information to our work. Sections III and IV present RTM and the hybrid traceability recovery method, respectively. Section V provides details on the design of the case study and presents the results achieved. Section VI discusses the results achieved, while Section VII concludes the paper.

II. Background

This section provides background notions and state of the art on IR-based traceability recovery.

A. IR-based Traceability Recovery

An IR-based traceability recovery tool uses an IR technique to compare a set of source artifacts (used as a query) against another (even overlapping) set of target artifacts and rank the similarities of all possible pairs of artifacts. The textual similarity between two artifacts is based on the occurrences of terms (words) within the artifacts contained in the repository. The extraction of the terms from the artifact contents is preceded by a text normalization for removing most non-textual tokens (e.g., operators, special symbols, some numbers) and splitting into separate words source code identifiers composed of two or more words separated by using the under_score or CamelCase separators. Common terms (e.g., articles, adverbs) that are not useful to capture semantics of the artifacts are also discarded using a stop word function, to prune out all the words having a length less than a fixed threshold, and a stop word list, to cut-off all the words contained in a given word list. In our study, we also performed a morphological analysis, i.e., stemming [11], of the extracted terms to remove suffixes of words to extract their stems.

The extracted information is generally stored in a $m \times n$ matrix (called *term-by-document* matrix), where m is the number of all terms that occur in all the artifacts, and n is the number of artifacts in the repository. A generic entry $w_{i,j}$ of this matrix denotes a measure of the weight (i.e., relevance) of the i^{th} term in the j^{th} document [3]. In our study we adopted a standard term weighting scheme known as *term frequency – inverse document frequency (td-idf)* [3]. Term frequency awards terms appearing in an artifact with a high frequency, while inverse document frequency penalizes terms appearing in too many artifacts, i.e., non-discriminating terms. This means that a term is considered relevant for representing the artifact content and is assigned a relatively high weight if it occurs many times in the artifact, and is contained in a small number of artifacts.

Based on the term-by-document matrix representation, different IR methods can be used to rank conceptual similarities between pairs of artifacts. In our study we use a probabilistic model, i.e., the JS model, VSM, and a topic model, i.e., RTM.

The JS similarity model is an IR technique driven by a probabilistic approach and hypothesis testing techniques. As well as other probabilistic models, it represents each artifact through a probability distribution. This means that an artifact is represented by a random variable where the probability of its states is given by the empirical distribution of the terms occurring in the artifact (i.e., normalized columns of the *term-by-document matrix*). The empirical distribution of a term is based on the weight assigned to the term for the specific artifact [10]. In the JS method the similarity between two artifacts is represented by the "distance" of their probability distributions measured by using the Jensen-Shannon Divergence [10]. The JS method does not take into account relations between terms. This means that having "automobile" in one artifact and "car" in another artifact does not contribute to the similarity measure between these two documents. Thus, the method suffers of the synonymy and the polysemy problems.

In the VSM, artifacts are represented as vectors of terms that occur within artifacts in the repository [3]. In particular, each column of the *term-by-document* matrix can be considered as an artifact vector in the m-space of the terms. Thus, the similarity between two artifacts is measured by the cosine of the angle between the corresponding vectors (i.e., columns of the *term-by-document* matrix). Such a similarity measure increases as more terms are shared between the two artifacts. In particular, as well as the JS method, VSM does not take into account relations between terms and it suffers of the synonymy and the polysemy problems.

Other than canonical IR-based recovery methods, we also propose the use of RTM as traceability recovery method. Details on such a technique are provided in Section III.

B. State of the art

Antoniol *et al.* [1] are the first to apply IR methods to the problem of recovering traceability links between software artifacts. They use both the probabilistic and vector space models to trace source code onto software documentation. The results of the experimentation show the two methods exhibit similar accuracy. Marcus and Maletic [5] use LSI to recover traceability links between source code and documentation. They perform case studies similar in design to those in [1] and compare the accuracy of LSI with respect to the vector space and probabilistic models. The results show that LSI performs at least as well as the probabilistic and vector space models combined with full parsing of the source code and morphological analysis of the documentation. Abadi *et al.* [10] compare several IR techniques to recover traceability links between code and documentation. They compare dimensionality reduction methods (e.g., LSI), probabilistic and information theoretic approaches (i.e., JS), and the standard VSM. The results achieved show that the techniques that provide the best results are VSM and JS. Recently, Asuncion *et al.* [8] applied LDA for traceability link recovery between text-based artifacts (such as requirements and design documents). The authors monitor the operations

(e.g., opening a requirements specification or visiting a Wiki page) performed by the software engineers during software development identifying a list of potentially related artifacts. Such relationships are then used to extract a set of topics that can be subsequently used to infer other relationships between code and documentation.

Heuristics [12], [13] and variants of basic IR methods [13], [14], [15], [16] have been proposed to improve the retrieval accuracy of IR-based traceability recovery tools. Promising results have also been achieved using the relevance feedback analysis [17], [18], [19] that aims at improving the accuracy of the tool by learning from user feedback provided during the link classification. Recently, the use of the coverage link analysis has also been proposed to increase the amount of correct links traced by the software engineer with respect to a traditional process [20].

A issue which hinders the performance of IR techniques when applied to traceability recovery is the presence of vocabulary mismatch between source and target artifacts. Recently, a technique attempts to alleviate such an issue has been introduced [21], [22]. The proposed approach uses search engines to identify a set of terms related to the query and expand the query in an attempt to improve recovery accuracy. Empirical studies indicate that using web mining to enhance queries improves retrieval accuracy.

III. Relational Topic Model

Relational Topic Model [9] is a hierarchical probabilistic model of links and document attributes. RTM defines a comprehensive method for modeling interconnected networks of documents. There exist other models for explaining network link structure (see related work by Chang *et al.* [9]), but what separates RTM from those prior methods of link prediction is its ability to account for both document context and links between documents when making predictions. Prediction of links, which are modeled as binary random variables, is dependent on the topic assignments of the documents modeled. Another distinction, beneficial to our application, is that RTM does not require any prior observed links to make these predictions.

Generating a model consist of two steps (1) modeling the documents in a corpus and (2) modeling the links between pairs of documents. Established with a foundation on LDA, step one is identical to the LDA generative process. In the context of LDA, each document has a corresponding multinomial distribution over T topics and each topic has a corresponding multinomial distribution over the set of words in the vocabulary of the corpus. LDA assumes the following generative process for each document d_i in a corpus D:

1) Choose $N \sim$ Poisson distribution (ξ)
2) Choose $\theta \sim$ Dirichlet distribution (α)
3) For each of the N words w_n:
   a) Choose a topic $t_n \sim$ Multinomial (θ).
   b) Choose a word w_n from $p(w_n|t_n, \beta)$, a multinomial probability conditioned on topic t_n.

The second phase for the generation of the model exploited by RTM is as follows:

For each pair of documents d_i, d_j:

a) Draw binary link indicator $y_{d_i,d_j}|t_i, t_j \sim \psi\left(\eta \cdot |t_i, t_j, \right)$ where $t_i = \{t_{i,1}, t_{i,2}, \ldots, t_{i,n}\}$

The link probability function ψ_ϵ is defined as:

$$\psi_\epsilon(y = 1) = \texttt{exp}(\eta^T(\overline{\texttt{t}}_{d_i} \circ \overline{\texttt{t}}_{d_j}) + v).$$

where links between documents are modeled by logistic regression. The $\circ$ notation represents the Hadamard product, $\overline{\texttt{t}}_d = \frac{1}{N_d}\sum_n z_{d,n}$ and `exp()` is an exponential mean function parameterized by coefficients η and intercept v.

Proposed applications of RTM [9] include assisting social network users in identifying potential friends, locating relevant citations for a given scientific paper, pinpointing related web pages of a particular web page, and computing coupling among source code classes in software [23]. Our intuition leads us to believe this model may serve well for traceability link recovery. In the context of traceability recovery, RTM is used to estimate topic distribution in the term-by-document matrix in order to define the link probability function. Such a function plays the same role of the artifact vectors in canonical vector-based IR methods, e.g., VSM. In particular, it is used to topically compare pairs of artifacts in order to obtain a list of candidate links.

One key distinction between establishing link probabilities in RTM and the canonical LDA is the underlying data used. Here, RTM uses topic assignments to make link predictions whereas to compute document similarities we use topic proportions for each document. This difference is discussed in more detail in the original work by Chang *et al.* [9].

IV. The Hybrid Approach

Besides proposing to use RTM as a traceability recovery method, we also propose a new approach to improve the accuracy of recovery methods by combining orthogonal IR methods, i.e. methods that provide different sets of recovered links. Our conjecture is supported by a preliminary study [6] that provides some evidence of (i) the equivalence (in terms of links recovered) of canonical IR methods, such as VSM, LSI, and JS and (ii) the presence of orthogonality between canonical IR methods and topic modeling techniques, in particular LDA. The proposed combined method is based on affine transformation [24], a technique used to combine experts' judgments previously used to combine orthogonal feature location techniques [25].

The basic idea behind our approach is that two IR methods can be viewed as two experts who provide their expertise to solve the problem of identifying links between a set of source artifacts and a set of target artifacts. The two experts, e.g., a canonical IR method and a topic modeling technique,

Table I
CHARACTERISTICS OF THE SOFTWARE SYSTEMS USED IN THE EXPERIMENTATION.

System	Description	Source Artifact (#)	Target Artifact (#)	Correct links
eAnci	A system providing support to manage Italian municipalities	Use cases (139)	Classes (55)	567
		Use cases (30)	Classes (37)	93
EasyClinic*	A system used to manage a doctor's office	UML Diagrams (20)	Classes (37)	69
		Test Cases (63)	Classes (37)	204
eTour*	An electronic touristic guide developed by students.	Use cases (58)	Classes (174)	366
SMOS	A system used to monitor high school students (e.g., absence, grades)	Use cases (67)	Classes (100)	1,044

* A complete version of the software system is available in both English (ENG) and Italian (ITA). Evaluation is performed on each version separately.

express their judgments based on different observations. Both experts express judgments based on the textual similarity between two artifacts. However, canonical methods analyze the terms shared by two artifacts, while topic modeling techniques utilize probabilistic topic distributions and word distributions across all the artifacts. This allows two techniques to capture different information, as highlighted in [6] and confirmed in our study (see Section V). Thus, the proposed approach combines valuable (orthogonal) expertise of both experts to obtain a more accurate list of candidate links and minimize the effort of software developers.

Formally, the combination is obtained in two steps. In the first step, the judgments (i.e., similarities) of the two experts are mapped to a standard normal distribution as follows:

$$sim_{m_i}(x,y) = \frac{m_i(x,y) - mean(m_i(X,Y))}{stdev(m_i(X,Y))}$$

where X, Y are sets of software artifacts, $x \in X, y \in Y$ and $sim_{m_i}(x,y)$ is the normalized similarity of $m_i(x,y)$ where m_i is an IR method. The functions $mean()$ and $stdev()$ return the mean and standard deviation respectively, for the similarity values of all pairs of artifacts (x_a, y_b) using m_i. Note that the normalization phase is required because different experts may express judgments that are not commensurable.

In the second step, the normalized judgments are combined through a weighted sum:

$$sim_{combined}(x,y) = \lambda \times sim_{m_i}(x,y) + (1-\lambda) \times sim_{m_j}(x,y)$$

where $\lambda \in [0,1]$ expresses the confidence in each technique. The higher the value the higher the confidence in the technique. In Section V-D we experimentally identify two heuristics to define the value of λ.

V. CASE STUDY

In this section we describe in detail the design and the results of the case study carried out to evaluate the proposed approach. The description of the study follows the Goal−Question−Metric [26] guidelines.

A. Definition and Context

The *goal* of the experiment was to analyze (i) the support given by RTM during traceability link recovery; (ii) whether RTM is orthogonal to VSM and JS canonical IR methods; and (iii) whether the accuracy of IR-based traceability recovery methods improves when combining RTM with other canonical methods. The *quality focus* was on ensuring better recovery accuracy, while the *perspective* was both (i) of a researcher, who wants to evaluate the accuracy improvement achieved using a hybrid recovery method; and (ii) of a project manager, who wants to evaluate the possibility of adopting the hybrid technique within her software company.

The context of our study is represented by six software repositories, namely eAnci, EasyClinic (English and Italian versions), eTour (English and Italian versions), and SMOS. All the systems have been developed by final year students at the University of Salerno (Italy). Use cases and code classes are available for eAnci, eTour, and SMOS, while for EasyClinic those two types of artifacts as well as UML interaction diagrams and test cases are available. Note that EasyClinic and eTour were recently used as data set for the traceability challenge organized at TEFSE 2009[2] and 2011[3].

Table I shows the characteristics of the considered software systems in terms of type and number of source and target artifacts. The language of the artifacts for all the systems is Italian, while for the EasyClinic and eTour repositories both Italian and English versions are available. On each system links between source and target artifacts are recovered to analyze the accuracy of the experimented IR methods. The table also reports the number of correct links between source and target artifacts. The traceability links were derived from the traceability matrix provided by the original developers. Such a matrix was used as the oracle for evaluating the accuracy of the studied traceability recovery methods.

B. Research Questions

In the context of our study the followings research questions (**RQ**) were formulated:

- **RQ**$_1$: *Does RTM-based traceability recovery outperform other canonical IR-based approaches?*
- **RQ**$_2$: *Is RTM orthogonal as compared to canonical IR techniques?*
- **RQ**$_3$: *Does the combination of RTM and canonical IR methods outperform stand-alone methods?*

To respond to our research questions, we recovered traceability links between source code and documentation of EAnci, EasyClinic, eTour, and SMOS (see Table I for details).

[2]http://web.soccerlab.polymtl.ca/tefse09
[3]http://www.cs.wm.edu/semeru/tefse2011

To have a good benchmark for the proposed traceability recovery methods and cover a large number of IR methods, we selected and considered as canonical method the JS method and VSM based on the results of our previous study [6]. The selected techniques are widely used for traceability recovery and are accepted as state of the art for IR-based traceability recovery [27]. In the context of our study, IR methods were provided identical term-by-document matrices as an input in order to eliminate all pre-processing related biases.

We were also interested in analyzing how the proposed approach interacts with the types and the language of the artifacts to be traced. Thus, two more research questions were formulated:

- **RQ**$_4$: *Does the type of the artifacts to be traced interact with the IR method and affect the recovery accuracy?*
- **RQ**$_5$: *Does the language of the artifacts to be traced interact with the IR method and affect the recovery accuracy?*

To analyze the effect of the type of the artifacts to be traced, only EasyClinic (English and Italian) repositories were considered because it is the only repository in our dataset with different types of artifacts. Regarding the influence of the language, we used both the EasyClinic and eTour repositories as for these repositories we had versions of the artifacts written in both Italian and English.

C. Metrics

To evaluate the accuracy of each IR method the number of correct links and false positives were collected for each recovery activity performed. Indeed, the number of correct links and false positives were automatically identified by a tool. The tool takes as an input the ranked list of candidate links and classifies each link as correct link or false positive until all correct links are recovered. Such a classification is automatically performed by the tool exploiting the original traceability matrix as an oracle.

Method comparison. A preliminary comparison of different IR methods—i.e., research questions RQ$_1$ and RQ$_3$—is obtained using two well-known IR metrics, namely recall and precision [3]:

$$recall = \frac{|cor \cap ret|}{|cor|}\% \qquad precision = \frac{|cor \cap ret|}{|ret|}\%$$

where cor and ret represent the sets of correct links and links retrieved by the tool, respectively. Other than recall and precision, we also use average precision [3], which returns a single value for each ranked lists of candidate links provided.

A further comparison of the IR-based recovery methods exploits statistical analysis. In particular, we used a statistical significance test to verify that the number of false positives retrieved by one method is significantly lower than the number of false positives retrieved by another method. In other words, we compared the false positives retrieved by method m_i with the false positives retrieved by method m_j to test the following null hypothesis:

> *H_0: there is no difference between the number of false positives retrieved by m_i and m_j*

Thus, the dependent variable of our study is represented by the number of false positives retrieved by the traceability recovery method for each correct link identified. Since the number of correct links is the same for each traceability recovery activity (i.e., the data was paired), we decided to use the Wilcoxon Rank Sum test [28] to test the statistical significance difference between the false positives retrieved by two traceability recovery methods. The results were intended as statistically significant at $\alpha = 0.05$.

Other than testing the null hypothesis, it is of practical interest to estimate the magnitude of the difference between accuracy achieved with different IR methods (e.g., combined *vs.* stand-alone). To this aim, we used the Cohen d effect size [29], which indicates the magnitude of the effect of the main treatment on the dependent variables [29]). For dependent samples (to be used in the context of paired analysis) it is defined as the difference between the means, divided by the standard deviation of the (paired) differences between samples, i.e., false positive distributions. The effect size is considered small for $0.2 \leq d < 0.5$, medium for $0.5 \leq d < 0.8$ and large for $d \geq 0.8$ [30]. We chose the Cohen d effect size as it is appropriate for our variables (in ratio scale) and given the different levels (small, medium, large) defined for it, it is quite easy to be interpreted.

Orthogonality Checking. To analyze the orthogonality of different IR methods (**RQ**$_2$), we uses Principal Component Analysis (PCA), a statistical technique capable of identifying various orthogonal dimensions captured by the data (principal components) and which measure contributes to the identified dimensions. The analysis identifies variables (in our case IR-based techniques) which are correlated to principal components and which techniques are the primary contributors to those components. This information provides insights on the orthogonality between similarity metrics.

Moreover, to have a further analysis of orthogonality between traceability recovery methods we used the following overlap metrics [6]:

$$correct_{m_i \cap m_j} = \frac{|correct_{m_i} \cap correct_{m_j}|}{|correct_{m_i} \cup correct_{m_j}|}\%$$

$$correct_{m_i \setminus m_j} = \frac{|correct_{m_i} \setminus correct_{m_j}|}{|correct_{m_i} \cup correct_{m_j}|}\%$$

where $correct_{m_i}$ represents the set of correct links identified by the IR method m_i. It is worth noting that $correct_{m_i \cap m_j}$ captures the overlap between the set of correct links retrieved by two IR methods, while $correct_{m_i \setminus m_j}$ measures the correct links retrieved by m_i and missed by m_j. The latter metric gives an indication on how an IR method contributes

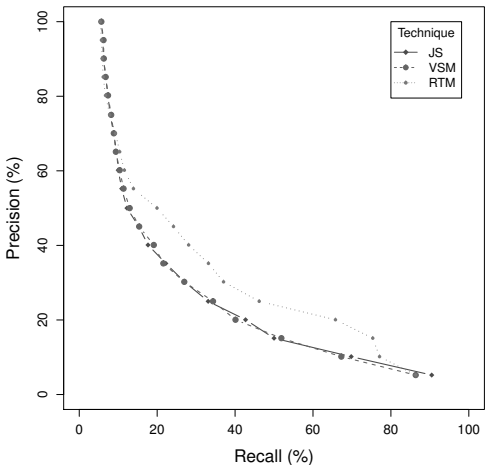


Figure 1. RTM vs VSM and JS: use cases onto code classes of eTour$_{ENG}$.

to complementing the set of correct links identified by the other method.

Interaction of Artifact Types and Language. The interaction of the type and the language of the artifacts to be traced with the IR method (**RQ**$_4$ and **RQ**$_5$) was analyzed by using the Two-Way Analysis of Variance (ANOVA) [31] and interaction plots. The latter are simple line graphs where the means on the dependent variable (number of false positives) for each level of one factor are plotted over all the levels of the second factor. When there is no interaction the resulting profiles are parallel, otherwise they are non-parallel [31].

D. Analysis of the Results

RQ$_1$: Accuracy of RTM. We first investigate whether RTM provides accuracy superior to that of other IR-based traceability recovery techniques. Fig. 1 provides the precision/recall curves achieved when tracing use cases onto code classes of eTour$_{ENG}$. For complete analysis see our online appendix [32] which, for each system, presents average precision for all IR-based techniques. From the results we are unable to identify an approach, which consistently exceeds the performance of all the others. As the figure shows, we have cases where RTM outperforms most other techniques for certain levels of recall, but there are also cases (e.g., on EasyClinic$_{ITA}$) where the performances of RTM are not consistently better than that acquired by other techniques.

The statistically analysis (see [32] for details) indicates that the RTM-based technique is capable of providing statistically significant improvement over other canonical techniques only when tracing use cases onto code classes and interaction diagrams onto code classes of eTour$_{ENG}$ (p-values are lower than 0.01 with a high effect size). In all the other cases, the Wilcoxon tests indicate that RTM does not provide any statistically significant improvement over other stand-alone methods.

RQ$_2$: Orthogonality checking. Regarding the orthogonality of the experimented techniques, Table II reports the results of PCA, which indicate the prevailing characteristics of the analysis (see [32] for all the results). Results of the six systems evaluated are in agreement with regards to both the number of principal components, which capture most of the variance, and the main contributors of those principal components. From the results, we can conclude that RTM is orthogonal to other canonical IR methods, that on the other hand are not orthogonal between them.

We also evaluated the degree of overlap amongst correct links for candidate sets provided by pairs of the techniques (one of the two techniques is the RTM-based traceability recovery technique). Given the top μ candidate links we describe two aspects of the data. Provided the set of correct links obtained from both candidate sets we determine the percentage of correct links (1) identified by both techniques ($correct_{RTM \cap JS}$) and (2) distinctly revealed by RTM ($correct_{RTM \setminus JS}$). Once again, Table III shows a subset of the results achieved (among the average results), while the complete analysis is reported in [32]. As we can see, the overlap between RTM and other techniques is relatively low, while the percentage of links identified by RTM and not identified by other canonical methods is high. In the case of recovering traceability links between use cases and code classes for EasyClinic$_{ENG}$ the results show that RTM provides a significant number of unique correct links. When ranked lists of 100 links are returned for the two techniques JS and RTM, JS is capable of identifying 33 correct links while RTM identifies 45 correct links. Among the correct links identified, 37% of them are common to both techniques while 42% are unique to RTM. Similar results are obtained for various systems, tracing links between different artifacts and for artifacts in various natural languages. These results confirm the findings of the PCA, indicating that RTM is a technique orthogonal to the other IR canonical methods.

RQ$_3$: Evaluation of the hybrid approach. Our goal is to improve traceability recovery accuracy by exploiting the orthogonality of IR methods. The proposed hybrid approach uses a parameter (λ) to assign a weight to the IR method to be combined (see Section IV). We analyze the effect of such a parameter on the accuracy (in terms of average precision) of the proposed approach using various values to lambda (0.05 through 0.95 with a step of 0.05) to combine techniques. Figure 2 shows the results achieved on ETour$_{ENG}$ (the complete analysis is reported in [32]). As expected the value of λ affects the accuracy of the proposed approach. Defining a "good" value for λ *a priori* is challenging. However, from the analysis of the results we identify two possible heuristics: (i) assign the same weight $\lambda = 0.5$ to the IR methods to be combined; (ii) use the proportion of variance obtained by PCA to weight the different IR methods. The former is a constant heuristic that generally provides good results, while the latter is an heuristic that is context-dependent and provides a more accurate estimation of λ. Such an heuristic is based on

Table II
PRINCIPAL COMPONENT ANALYSIS. RESULTS ARE FOR TRACING USE CASES ONTO CODE CLASSES.

	PC_1	PC_2	PC_3
% variance	79.74%	20.15%	0.11%
Cumulative %	79.74%	99.89%	100%
JS	**0.98**	-0.19	0.03
VSM	**0.97**	-0.19	-0.03
RTM	0.68	**0.73**	0.00

(a) EasyClinic-ENG

	PC_1	PC_2	PC_3
% variance	75.78%	24.11%	0.11%
Cumulative %	75.78%	99.89%	100%
JS	**0.99**	-0.09	0.04
VSM	**0.99**	-0.10	-0.03
RTM	0.54	**0.83**	0.00

(b) EasyClinic-ITA

	PC_1	PC_2	PC_3
% variance	68.51%	31.18%	0.31%
Cumulative %	68.51%	99.69%	100%
JS	**0.99**	0.11	0.07
VSM	**0.99**	0.08	-0.06
RTM	0.29	**0.95**	0.00

(c) eTour-ENG

	PC_1	PC_2	PC_3
% variance	67.12%	32.63%	0.25%
Cumulative %	67.12%	99.75%	100%
JS	**0.97**	0.21	0.06
VSM	**0.98**	0.18	-0.05
RTM	0.31	**0.94**	0.00

(d) eTour-ITA

	PC_1	PC_2	PC_3
% variance	63.79%	35.55%	0.66%
Cumulative %	63.79%	99.34%	100%
JS	**0.96**	0.24	0.10
VSM	**0.97**	0.18	-0.09
RTM	0.17	**0.98**	0.00

(e) SMOS

	PC_1	PC_2	PC_3
% variance	70.03%	29.65%	0.32%
Cumulative %	70.03%	99.68%	100%
JS	**0.98**	-0.19	-0.06
VSM	**0.98**	-0.18	0.06
RTM	0.42	**0.90**	0.00

(f) EAnci

Table III
OVERLAP ANALYSIS. RESULTS ARE FOR TRACING USE CASES ONTO CODE CLASSES.

	EasyClinic$_{ITA}$			EasyClinic$_{ENG}$			eTour$_{ENG}$			eTour$_{ITA}$			EAnci			SMOS		
	Cut points μ			Cut points μ			Cut points μ			Cut points μ			Cut points μ			Cut points μ		
	25	50	100	25	50	100	25	50	100	25	50	100	25	50	100	25	50	100
$correct_{JS \cap VSM}$	100	92	95	83	85	88	91	94	85	89	91	91	72	80	82	100	79	76
$correct_{JS \setminus VSM}$	0	3	4	16	15	8	4	5	7	10	5	7	11	7	10	0	8	6
$correct_{VSM \setminus JS}$	0	3	0	0	0	2	4	0	7	0	2	1	16	11	8	0	12	17
$correct_{JS \cap RTM}$	19	40	52	19	19	36	23	28	35	25	36	36	20	23	29	16	18	23
$correct_{JS \setminus RTM}$	42	22	21	38	36	21	41	35	22	34	29	25	41	26	24	26	24	23
$correct_{RTM \setminus JS}$	38	37	26	42	44	42	35	36	41	40	34	38	37	50	45	56	57	52
$correct_{VSM \cap RTM}$	19	40	51	15	17	33	23	29	32	26	34	35	15	26	31	16	17	26
$correct_{VSM \setminus RTM}$	42	22	20	35	32	21	41	32	24	30	29	23	46	26	22	26	25	25
$correct_{RTM \setminus VSM}$	38	37	28	50	50	45	35	38	42	43	36	40	38	47	45	56	56	47

Table IV
COMPARING RTM-BASED COMBINATIONS WITH STAND-ALONE METHODS: WILCOXON TEST RESULTS (P-VALUES).

	EasyClinic$_{ENG}$	EasyClinic$_{ITA}$	eTour$_{ENG}$	eTour$_{ITA}$	SMOS	EAnci
RTM+JS vs JS	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	1	$<$ **0.001**
RTM+VSM vs VSM	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	1	$<$ **0.001**
RTM+JS vs RTM	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	0.47
RTM+VSM vs RTM	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	**0.04**

the observation that PCA identifies the different dimensions that describe a phenomenon, e.g., the similarity between pairs of artifacts, and gives an indication of the importance of each dimension (captured by one or more IR methods) in the description of this phenomenon, i.e., the proportion of variance. We conjecture that the higher the amount of variance captured by a particular dimension the higher should be the weight for the IR technique that best correlates to that dimension. The accuracy obtained weighting the IR methods exploiting the proportion of variance obtained by PCA is highlighted in Figure 2 with an asterisk. Note that the PCA-based weighting technique provides better results than approximately 75% of combinations considered (see [32] for the complete analysis). Such a result suggests that the PCA-based technique provides an acceptable means of combining IR methods for recovering traceability links.

Figure 2 also highlights the benefits provided by combining orthogonal IR methods. In particular, the accuracy of RTM+JS (or VSM) sensibly overcomes the accuracy of JS+VSM. To have further evidence of the benefits provided by the combination of different IR methods (using for $lambda$ the best and the PCA-based values), Figure 3 shows the average precision achieved with stand-alone methods and different combinations of IR methods. As we can see the combination of RTM with other IR techniques results in significant improvement in average precision. In addition, the results achieved applying our proposed PCA-based weighting technique to combine orthogonal IR methods yields results, which consistently exceed the results of standalone techniques. Such a result confirms the usefulness of the proposed heuristic.

All these findings were also confirmed by the results of the Wilcoxon tests (see Table IV). In all the repositories, but SMOS and in one case for EAnci, the RTM combined method is able to statistically outperform the stand-alone methods. However, even if in the other cases the results did not reveal a statistically significant difference between techniques, the average precision of the combination is higher than any other standalone method.

RQ$_{4,5}$: Interaction of Artifact Type and Language. The ANOVA analysis confirmed the influence of the IR method, and highlighted the influence of both types and language of the artifacts to be traced. ANOVA also revealed a statistically significant interaction between IR method and artifact language (on the ETour repositories), as well as between IR method and artifact type. The interactions

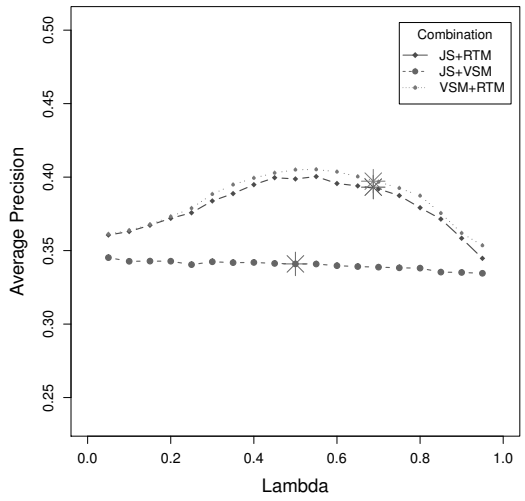


Figure 2. Average precision in eTour$_{ENG}$ using various values of lambda. Lambda represents the weight of the first method in the combination, while the asterisk indicates the accuracy of the PCA-based weighting technique.

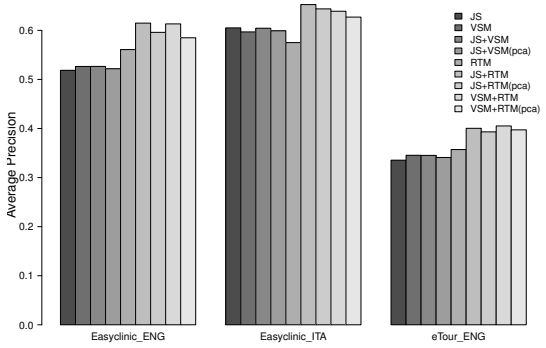


Figure 3. Results of average precision for retrieving all correct links for each EasyClinic$_{ENG}$ (left), EasyClinic$_{ITA}$ (middle), and eTour$_{ENG}$ (right). Results are presented the best performing combination and combinations obtained using the PCA-based weighting technique.

investigated are statistically significant based on our dataset with $p < 0.001$. To better understand the interaction between factors, Figure 4 shows (a) the interaction plot between IR method and artifact language and (b) between IR method and artifact type. Regarding the influence of the artifact language, we observe that on EasyClinic better recovery accuracy is achieved on the Italian version, while on ETour better accuracy is generally achieved on the English version. The reason is that in the Italian version of the ETour repository identifiers in the source code are written in English. This negatively impacts the accuracy of the IR methods. As for the influence of the artifact type, we observe that the combination is highly valuable when tracing UML diagrams onto source code, while in the other cases the improvement is not so evident.

VI. Discussion and Threats to Validity

This section discusses the achieved results focusing the attention on the threats that could affect their validity [26].

A. Evaluation Method

Recall, precision, and average precision are widely used metrics for assessing an IR technique and the number of false positives retrieved by a traceability recovery tool for each correct link retrieved reflects well its retrieval accuracy. The overlap metrics give a good indication on the overlap of the correct links recovered by the different IR methods. Moreover, the similarity measures provided by each IR method are also statistically analyzed using PCA to verify the presence of IR methods that provide orthogonal similarity measures.

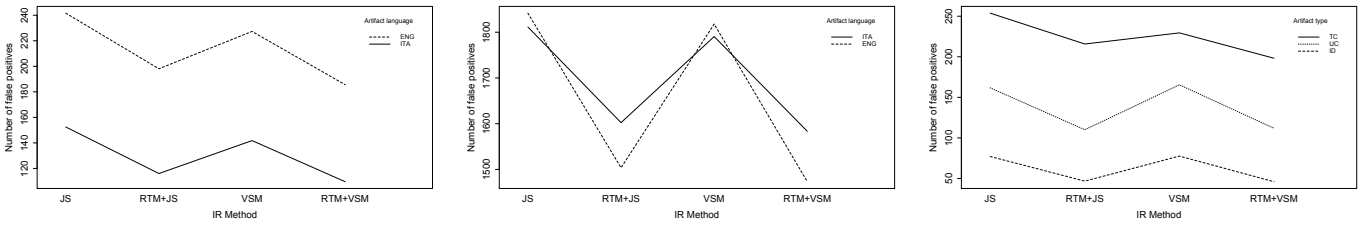
We also performed statistical analysis of the achieved results. Attention was paid not to violate assumptions made by statistical tests. Whenever conditions necessary to use parametric statistics did not hold (e.g., analysis of each experiment data), we used non-parametric tests, in particular Wilcoxon test for paired analysis. We also used a parametric test, i.e., ANOVA, to analyze the effect of different factors even if the distribution was not normal. According to [33] this can be done since the ANOVA test is a very robust test. In addition, even if the distribution is not normally distributed we can relax the normality assumption applying the law of large numbers. In particular, according to [34] having a population higher than 100 it is possible to safety relax the normality assumption.

B. Object Systems and Oracle Accuracy

An important threat is related to the repositories used in the case study. EAnci, EasyClinic, eTour, and SMOS are not industrial projects, since they were developed by students. However, they are comparable to (or greater than) repositories used by other researchers [7], [1], [19], [35], [36] and both EasyClinic and ETour have been used as benchmark repositories in the last two editions of the traceability recovery challenge organized at TEFSE. In addition, to the best of our knowledge in this paper we reported the largest empirical study to evaluate and compare different IR methods for traceability recovery.

The investigated traceability recovery methods are based on IR techniques. Thus, the language of the artifacts may play an important role and affect the achieved results. To mitigate such a threat we performed the experimentation on two versions of the same artifact repository, one written in Italian and the other one in English. Analyzing the performance achieved on the same repository written in two different natural languages we had possibility to focus our investigation on the only difference between two versions of the repository, i.e., artifact language. The same considerations hold for the types of the artifacts to be traced.

Finally, the accuracy of the oracle we used to evaluate the tracing accuracy could also affect the achieved results. To mitigate such a threat we used original traceability matrices provided by the software developers. The links were also validated during review meetings made by the



(a) EasyClinic - Method and Language (b) eTour - Method and Language (c) EasyClinic - Method and Artifact Type

Figure 4. Interaction between Method and Artifact Types and between Method and Language.

original development team together with PhD students and academic researchers.

C. RTM Configuration and Number of Topics

RTM is a probabilistic topic model method which uses sampling techniques to infer underlying topics and topic/word distributions. When generating topic models, using an R project implementation[4], we performed a large number of sampling iterations to stabilize the set of topics extracted from a software system. In addition, the choice of the number of topics is critical and the proper way to make such a choice is still an open issue. For this reason we experimented different number of topics and for each repository we used the value that provides the best accuracy. Future work will be devoted to try to identify an heuristic to estimate the number of topics.

D. Heuristics to Weight the IR methods to be Combined

The proposed hybrid approach uses a parameter (λ) to assign a weight to the IR method to be combined. Defining a "good" value for λ *a priori* is challenging. For this reason, we experimentally identified two possible heuristics to weight the IR methods to be combined: (i) assign the same weight $\lambda = 0.5$ to the IR methods to be combined; (ii) use the proportion of variance obtained by PCA to weight the different IR methods. Both the heuristics are able to approximate the optimal λ. This means that the software engineer can initially use the value provided by the heuristic and then work around it by slightly increasing or decreasing it, within an incremental classification process.

E. Orthogonality is a Key Point for Improving Accuracy

In our study we compare the accuracy of different IR methods, namely RTM, JS, and VSM. No IR method consistently provides superior recovery accuracy when compared to all other IR-based techniques considered. In particular, there are several cases where applying different IR-based traceability recovery techniques result in comparable accuracy and there are also cases where a particular technique yields better accuracy than any other technique considered.

[4]http://cran.r-project.org/web/packages/lda/

The results achieved also highlight that JS and VSM are almost equivalent, while RTM captures a unique dimension in the data, i.e., it identifies correct links overlooked by JS and VSM. Across all systems evaluated, PCA reveals that there exists a principal component (typically accounting for 20%-35% of variance in data) with RTM as its main contributor. That is, RTM tends to contribute 73%-98% of the variance captured by that particular principal component. Through our analysis of overlap of links between pairs of IR methods we confirm that RTM is able to provide correct links omitted by other techniques for particular cut points.

Orthogonality of IR-based techniques is a key point for improving accuracy through combining different techniques. In our study we show that the combination of RTM with orthogonal IR techniques results in accuracy which surpasses that of either stand-alone technique. That is, in our results the improvements in precision exceed 30% in certain cases. Although improvements of that magnitude do not occur across all the systems evaluated, we do obtain acceptable increases in virtually all the scenarios.

VII. Conclusion

In this paper we presented a novel traceability recovery method based on RTM and an hybrid approach for traceability recovery that combines different IR methods. We also analyzed (i) the orthogonality of RTM as compared to other IR methods and (ii) the recovery accuracy improvement provided by the combination of RTM with other canonical IR methods. The empirical case study conducted on six software systems indicated that the hybrid method outperforms stand-alone IR methods as well as any other combination of non-orthogonal methods with a statistically significant margin.

Future work will be devoted to replicating our study to corroborate these findings. Moreover, there are a number of directions on how to improve the accuracy of the proposed traceability recovery methods. A first direction aims at defining a more sophisticated method for combining RTM with other IR methods, including utilizing structural information [37]. A second direction aims at integrating a specialized learning algorithm exploiting the relevance feedback analysis into the approach.

ACKNOWLEDGMENTS

We thank ICSM'11 reviewers for pertinent comments, which helped us to improve the quality of the paper. This work is supported in part by NSF CCF-1016868, NSF CCF-0916260, and NSF CNS-0959924 awards to the College of William and Mary. Any opinions, findings and conclusions expressed herein are the authors and do not necessarily reflect those of the sponsors.

REFERENCES

[1] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo, "Recovering traceability links between code and documentation," *IEEE TSE*, vol. 28, no. 10, 2002.

[2] B. Ramesh and M. Jarke, "Toward reference models for requirements traceability," *IEEE TSE*, vol. 27, 2001.

[3] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, 1999.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[5] A. Marcus, J. I. Maletic, and A. Sergeyev, "Recovery of traceability links between software documentation and source code," *Journal of Software Engineering and Knowledge Engineering*, vol. 15, no. 5, pp. 811–836, 2005.

[6] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia, "On the equivalence of information retrieval methods for automated traceability link recovery," in *Proc. of ICPC*, 2010.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[8] H. U. Asuncion, A. Asuncion, and R. N. Taylor, "Software traceability with topic modeling," in *Proc. of ICSE*, 2010.

[9] J. Chang and D. M. Blei, "Hierarchical relational models for document networks," *Annals of Applied Statistics*, 2010.

[10] A. Abadi, M. Nisenson, and Y. Simionovici, "A traceability technique for specifications," in *Proc. of ICPC*, 2008.

[11] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[12] G. Capobianco, A. De Lucia, R. Oliveto, A. Panichella, and S. Panichella, "On the role of the nouns in ir-based traceability recovery," in *Proc. of ICPC*, 2009.

[13] J. Cleland-Huang, R. Settimi, C. Duan, and X. Zou, "Utilizing supporting evidence to improve dynamic requirements traceability," in *Proc. of RE*, 2005, pp. 135–144.

[14] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora, "Recovering traceability links in software artefact management systems using information retrieval methods," *ACM TOSEM*, vol. 16, no. 4, 2007.

[15] M. Lormans, A. Deursen, and H.-G. Gross, "An industrial case study in reconstructing requirements views," *Empirical Soft. Eng.*, vol. 13, no. 6, pp. 727–760, 2008.

[16] R. Settimi, J. Cleland-Huang, O. Ben Khadra, J. Mody, W. Lukasik, and C. De Palma, "Supporting software evolution through dynamically retrieving traces to UML artifacts," in *Proc. of IWPSE*, 2004, pp. 49–54.

[17] G. Antoniol, G. Casazza, and A. Cimitile, "Traceability recovery by modelling programmer behaviour," in *Proc. of WCRE*, 2000, pp. 240–247.

[18] A. De Lucia, R. Oliveto, and P. Sgueglia, "Incremental approach and user feedbacks: a silver bullet for traceability recovery," in *Proc. of ICSM*, 2006, pp. 299–309.

[19] J. H. Hayes, A. Dekhtyar, and S. K. Sundaram, "Advancing candidate link generation for requirements tracing: The study of methods." *IEEE TSE*, vol. 32, no. 1, pp. 4–19, 2006.

[20] A. De Lucia, R. Oliveto, and G. Tortora, "The role of the coverage analysis in traceability recovery process: a controlled experiment," in *Proc. of ICSM*, 2009.

[21] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A machine learning approach for tracing regulatory codes to product specific requirements," in *Proc. of ICSE*, 2010, pp. 155–164.

[22] M. Gibiec, A. Czauderna, and J. Cleland-Huang, "Towards mining replacement queries for hard-to-retrieve traces," in *Proc. of ASE*, 2010, pp. 245–254.

[23] M. Gethers and D. Poshyvanyk, "Using relational topic models to capture coupling among classes in object-oriented software systems," in *Proc. of ICSM*, 2010.

[24] R. A. Jacobs, "Methods for combining experts' probability assessments," *Neural Computation*, vol. 7, no. 5, 1995.

[25] D. Poshyvanyk, Y. Gael-Gueheneuc, A. Marcus, G. Antoniol, and V. Rajlich, "Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval," *IEEE TSE*, vol. 33, no. 6, pp. 420–432, 2007.

[26] V. Basili, G. Caldiera, and D. H. Rombach, *The Goal Question Metric Paradigm*, 1994.

[27] J. Cleland-Huang, B. Berenbach, S. Clark, R. Settimi, and E. Romanova, "Best practices for automated traceability," *IEEE Computer*, vol. 40, no. 6, pp. 27–35, 2007.

[28] W. J. Conover, *Practical Nonparametric Statistics*, 3rd ed., 1998.

[29] V. B. Kampenes, T. Dybå, J. E. Hannay, and D. I. K. Sjøberg, "A systematic review of effect size in software engineering experiments." *Information and Software Technology*, vol. 49, no. 11-12, pp. 1073–1086, 2007.

[30] J. Cohen, *Statistical power analysis for the behavioral sciences*, 1988.

[31] J. L. Devore and N. Farnum, *Applied Statistics for Engineers and Scientists*, 1999.

[32] M. Gethers, R. Oliveto, D. Poshyvanyk, and A. D. Lucia, "On integrating orthogonal information retrieval methods to improve traceability recovery," The College of Williams and Mary, http://www.cs.wm.edu/semeru/data/icsm2011-traceability-rtm, Tech. Rep., 2011.

[33] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in Software Engineering - An Introduction*, 2000.

[34] R. M. Sirkin, *Statistics for the social sciences*. California, USA: Sage Publications, 2005.

[35] M. Lormans and A. van Deursen, "Can LSI help reconstructing requirements traceability in design and test?" in *Proc. of 10th European Conference on Software Maintenance and Reengineering*, Bari, Italy, 2006, pp. 45–54.

[36] A. Marcus and J. I. Maletic, "Recovering documentation-to-source-code traceability links using latent semantic indexing," in *Proc. of 25th International Conference on Software Engineering*, Portland, Oregon, USA, 2003, pp. 125–135.

[37] C. McMillan, D. Poshyvanyk, and M. Revelle, "Combining textual and structural analysis of software artifacts for traceability link recovery," in *Proc. of TEFSE*, 2009, pp. 41–48.