On the Equivalence of Information Retrieval Methods for Automated Traceability Link Recovery

Rocco Oliveto∗, Malcom Gethers†, Denys Poshyvanyk†, Andrea De Lucia∗

∗Department of Mathematics and Informatics, University of Salerno, via Ponte don Mellilo, Fisciano (SA), Italy
†Computer Science Department, The College of William and Mary, Williamsburg, VA 23185, USA
rolliveto@unisa.it, mgethers@cs.wm.edu, denys@cs.wm.edu, adelucia@unisa.it

Abstract—Different IR methods have been proposed for recovering traceability links between code and documentation. So far, there is no clear winner among the exploited IR techniques. In this paper we present an empirical study aiming at statistically analyzing the equivalence of several IR-based traceability recovery methods. The analysis is based on Principal Component Analysis and on the analysis of the overlap of the set of candidate links provided by each method. The studied techniques are three widely used IR methods – i.e., the Jensen-Shannon (JS) method, Vector Space Model (VSM), and Latent Semantic Indexing (LSI) – and Latent Dirichlet Allocation (LDA), an IR method previously used to support other software engineering tasks but never used for traceability recovery. The results show that while the three methods previously used for traceability recovery are equivalent, LDA is able to capture a dimension unique to the set of techniques which we considered. Moreover, although the accuracy of LDA is lower than previously used methods, in several cases the combination of LDA with other IR methods improves the traceability recovery accuracy of stand-alone methods.

Keywords—Traceability Recovery; Vector Space Model; Latent Semantic Indexing; Jensen-Shannon method; Latent Dirichlet Allocation; Empirical Studies.

I. INTRODUCTION

Maintaining dependencies (traceability links) between different types of software artifacts is widely recognized as an important support activity both during initial system development and also during the ongoing change management process. In particular, traceability links between the free text documentation associated with the development and maintenance cycle of a software system and its source code is helpful in a number of tasks – such as requirement coverage, program comprehension and impact analysis. Software artifact traceability is also considered as a “best practice” by numerous major software engineering standards (such as CMMI or ISO 15504).

Unfortunately, establishing and maintaining traceability links between software artifacts is a time consuming, error prone, and person-power intensive task. Consequently, despite the advantages that can be gained, effective traceability is rarely established unless there is a regulatory reason for doing so. Extensive effort in the software engineering community (both research and commercial) has been brought forth to improve the explicit connection of documentation and source code. Promising results have been achieved using Information Retrieval (IR) techniques [1], [2] for traceability recovery [3], [4], [5], [6], [7], [8], [9], [10], [11]. IR-based methods propose a list of candidate traceability links on the basis of the similarity between the text contained in the software artifacts. Such methods are based on the conjecture that two artifacts having high textual similarity share several concepts thus they are good candidates to be traced on each other. Several IR methods have been proposed for traceability recovery – e.g., vector space and probabilistic models [1] or Latent Semantic Indexing (LSI) [2]. In general, the retrieval accuracy of IR-based traceability recovery methods is assessed through two measures: recall, measuring the percentage of correct links that were found, and precision, measuring the percentage of found links that were correct. The results achieved are sometimes contrasting and demonstrate no clear winner among IR techniques. Indeed, it seems that all the exploited techniques so far are able to capture the same information when used to calculate the textual similarity between software artifacts.

In this paper we present an empirical study aiming at statistically analyzing the equivalence of different IR-based traceability recovery methods. The comparison is based on Principal Component Analysis (PCA) and on the analysis of the overlap of the set of candidate links provided by each of the IR methods. The studied IR techniques are the Jensen-Shannon (JS) method [3], the Vector Space Model (VSM) [1], LSI [2], and Latent Dirichlet Allocation (LDA) [12]. The first three methods were selected because they are widely used and seem to be the techniques that give the best results [3], [4], [10]. LDA was never used for traceability recovery. However, we also analyze the support given by such a technique during traceability recovery because in a previous study [13] the authors demonstrate that LDA is able to capture some aspects missed by other IR methods, such as LSI, when it is used to calculate the conceptual cohesion of a class.

The empirical analysis has been conducted on a relatively small software repository, i.e., EasyClinic, and on a larger repository, i.e., eTour. The studied IR methods have been used to recover traceability links between the use cases and
the source code of the two software systems. The results achieved demonstrate that JS, VSM, and LSI are almost equivalent, while LDA is able to capture some information missed by the other exploited IR methods. The analysis of recall and precision also demonstrate that the accuracy of LDA is lower than the accuracy of JS, VSM, and LSI. These considerations suggest that probably LDA can be used as a method to augment stand-alone methods – e.g., JS, VSM, and LSI – aiming at improving their accuracy. For this reason we present an approach for combining LDA with other exploited IR methods. The results achieved with the combined method seem quite promising. In particular, in several cases the combined method is able to overcome the stand-alone IR methods. However, the combination of LDA with other IR methods should be further explored aiming at sensibly improving the accuracy of the stand-alone methods. Summarizing, the specific contributions of the paper can be outlined as follows:

- an empirical study aiming at analyzing the equivalence of the support provided by different IR methods for traceability link recovery. To the best of our knowledge this is the first paper where several IR methods are empirically compared – not only using recall and precision – for analyzing the equivalence of the similarity measures they provide when used for identifying traceability links between software artifacts of established datasets;
- the definition of a LDA-based approach for recovering traceability links between software artifacts. LDA was previously used to support other software engineering tasks but never used for traceability recovery;
- a preliminary analysis of the combination of LDA with other IR methods aiming at improving the recovery accuracy of the stand-alone techniques.

The rest of the paper is organized as follows. Section II discusses related work, while Section III presents the IR-based traceability recovery process and briefly describes the different IR methods exploited in our study. Section IV provides details on the design of the case study. Sections V and VI report and discuss on the results achieved, respectively, while Section VII gives concluding remarks.

II. RELATED WORK

The use of IR methods for traceability recovery is introduced by Antoniol et al. [4]. They apply the probabilistic and vector space models [1] to trace source code onto software documentation. The comparison of the two techniques is based on recall and precision. The results achieved demonstrate that both models recover all correct links with almost the same number of links retrieved.

Marcus and Maletic [10] perform the same case studies as [4] and compare the accuracy of LSI [2] with respect to the vector space and probabilistic models. Also in this case the comparison is based on recall and precision. 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.

LSI and VSM are also compared in [7], [8] and the results achieved are quite in contrast with the results achieved in [10]. In particular, the analysis of recall and precision achieved by LSI and VSM do not highlight any method performing better than the other. The authors also try to improve the accuracy of both methods using the relevance feedback analysis [7]. Also in this case the results achieved do not highlight any techniques that better support the proposed enhancing strategy.

Besides relevance feedback analysis other variants of basic IR methods have been proposed to improve the retrieval accuracy of IR-based traceability recovery tools [5], [8], [9], [11]. However, the proposed variants are not able to sensibly overcome the accuracy of canonical methods especially when the goal is to recover all correct links. In particular, it is likely that there is an upper bound to the accuracy improvements that cannot be overcome, even if different enhancing strategies are used.

Recently, Abadi et al. [3] compared several IR techniques, including VSM, LSI, and the JS method, using recall and precision, to recover traceability links between code and documentation. Also in this case the achieved results are in contrast with the results achieved by Marcus and Maletic [10]. In particular, they demonstrate that LSI performs noticeably worse than VSM and that the techniques that provide the best results are the standard VSM and the JS method. However, the case study is conducted on a very small dataset containing less than ten classes and few requirements, which does not permit statistical generalization of the results.

Capobianco et al. [14] propose a novel IR method for traceability recovery that models the information contained in a software artefact by particular interpolation curves of plots mapping terms and their frequency on the artefact. Then, the similarity between artefacts is computed by calculating the distance of the corresponding interpolation curves. The results of a reported case study demonstrate that the proposed approach significantly outperforms VSM and LSI, and it is comparable and sometimes better than the JS method.

All these case studies compare different IR-based traceability recovery approaches using recall and precision. The results achieved do not highlight any clear winner among the studied IR methods. Indeed, it seems that all the exploited techniques are able to capture almost the same information. However, to the best of our knowledge there is no empirical study aiming at analyzing the equivalence of different IR-based traceability recovery approaches. For this reason, in our study other than analyzing the recall and precision achieved by the exploited methods, we also perform a
comparison based on PCA and on the analysis of the overlap of the set of candidate links provided by each method.

III. IR-BASED TRACEABILITY RECOVERY

IR methods index the documents in a document space as well as the queries by extracting information about the occurrences of terms within them. This information is used to define similarity measures between queries and documents. In the case of traceability recovery, this similarity measure is used to identify that a traceability link might exist between two artifacts, one of which is used as a query.

The construction of the vocabulary and the indexing of the artifacts are preceded by a text normalization phase generally performed in two steps (see Figure 1). In the first step white spaces and most non-textual tokens (e.g., special symbols, numbers) are pruned out from the text. Moreover, all capital letters are transformed into lower case letters. In the second step a stop word function and a stop word list are applied to discard common words (e.g., articles, adverbs) that are not useful to characterize the semantics of the artifact contents. In particular, the stop word function prunes out all the words having a length less than a fixed threshold (we fixed such a threshold to three letters [1]), while the stop word list is used to cut-off all the words contained in a given word list. In our study we used a stop word function in conjunction with a stop word list aiming at reducing the size of the stop word list [1]. In this way we also reduce the time required for term filtering. A morphological analysis, i.e., stemming [15], of the extracted terms can also be performed aiming at removing suffixes of words to extract their stems. However, in our study we did not use stemming.

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 within the artifacts, and $n$ is the number of artifacts in the repository. A generic entry $\alpha_{i,j}$ of this matrix denotes a measure of the weight (i.e., relevance) of the $i^{th}$ term in the $j^{th}$ document [1]. Different measures based on the frequency of the terms in the artifacts have been proposed for this weight. In our study we used the normalized term frequency [1].

The term-by-document matrix represents the input for the classifier. The classifier is in charge to compare a set of artifacts (e.g., use cases) against another set of artifacts (e.g., code classes) and rank the similarity of all possible pairs of artifacts. Then a threshold (e.g., a cut point [4]) is used to select the first $\mu$ documents in the ranked list. The tracing process is semiautomatic. Unfortunately, any IR method will fail to retrieve some of the correct links, while on the other hand it will also retrieve links between artifacts that are not correct (false positives).

Based on the term-by-document matrix representation, different IR methods can be used to rank the similarity between the pairs of artifacts. In our study we used a probabilistic method, i.e., the JS model [3], a vector space-based method, i.e., the VSM [1], a space reduction-based method, i.e., LSI [2], and a topic modeling method, i.e., LDA [12]. These methods are described in the following subsections.

A. The Jensen-Shannon Model

The JS similarity model is an IR technique recently proposed by Abadi et al. [3]. It is 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 matrix1). The empirical distribution of a term is based on the weight assigned to the term for the specific artifact [3].

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 [3]:

$$sim_{i,j} = 1 - \left( H \left( \frac{p_i + p_j}{2} \right) - \frac{H(p_i) + H(p_j)}{2} \right)$$

where $p_i$ and $p_j$ are the empirical distributions of the documents $i$ and $j$, respectively, and $H(p)$ is the entropy of distribution $p$ [16]:

$$H(p) = \sum_{j=1}^{n} p(x_j) \cdot \log p(x_j)$$

The JS method does not take into account relations between terms. This means that having “automobile” in one artifact

1To represent a probability distribution, each row of the term-by-document matrix has to be normalized aiming at having that the sum of the elements is equals to 1 [3].
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.

B. Vector Space Model

In the VSM, artifacts are represented as vectors of terms that occur within artifacts in the repository [1]. 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):

\[
sim_{i,j} = \frac{\sum_{k=1}^{m} a_{i,k} a_{j,k}}{\sqrt{\sum_{h=1}^{m} a_{i,h}^2} \cdot \sqrt{\sum_{l=1}^{m} a_{j,l}^2}}
\]

where \( a_{i,j} \) is a generic entry of the term-by-document matrix. It is worth noting that 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.

C. Latent Semantic Indexing

LSI [2] is an extension of the VSM developed to overcome the synonymy and polysemy problems. In particular, LSI is able to exploit information about co-occurrence of terms (latent structure) to automatically discover synonymy between different terms. For example, both “car” and “automobile” are likely to co-occur in different artifacts with related terms, such as “motor”, “wheel”, etc. In the same way the method also mitigates the polysemy problem.

To mitigate synonymy and polysemy problems, LSI decomposes the original term-by-document matrix into the product of three matrices by Singular Value Decomposition (SVD) [17]:

\[
A = T_0 \cdot S_0 \cdot D_0
\]

where \( T_0 \) is the \( m \times r \) matrix of the terms containing the left singular vectors (rows of the matrix), \( D_0 \) is the \( r \times n \) matrix of the documents containing the right singular vectors (columns of the matrix), \( S_0 \) is an \( r \times r \) diagonal matrix of singular values, and \( r \) is the rank of \( A \).

SVD can be viewed as a technique for deriving a set of uncorrelated indexing factors or concepts, whose number is given by the rank \( r \) of the matrix \( A \), and whose relevance is given by the singular values in the matrix \( S_0 \). Moreover, SVD allows a simple strategy for optimal approximate fit using smaller matrices. If the singular values in \( S_0 \) are ordered by size, the first \( k \) largest values may be kept and the remaining smaller ones set to zero. Since zeros were introduced into \( S_0 \), the representation can be simplified by deleting the zero rows and columns of \( S_0 \) to obtain a new diagonal matrix \( S \), and deleting the corresponding columns of \( T_0 \) and rows of \( D_0 \), to obtain \( T \) and \( D \) respectively. The result is a reduced model:

\[
A \approx A_k = T \cdot S \cdot D
\]

where the matrix \( A_k \) is only approximately equal to \( A \) and is of rank \( k < r \). The reduced matrix captures most of the important underlying structure in the association of terms and documents, yet at the same time it removes the noise or variability in word usage that plagues word-based retrieval methods. Intuitively, since the number of dimensions \( k \) is much smaller than the number of unique terms \( m \), minor differences in terminology will be ignored. In the application of LSI to canonical IR activities, good performances have been achieved using about 100 concepts on a document space of about 1,000 documents and a vocabulary of about 6,000 terms [2]. In our study we also set \( k = 100 \).

The matrix \( S \cdot D \) projects the document vectors from the \( m \)-space of the terms to the reduced \( k \)-space of concepts. The cosine of the angle between two vectors in this \( k \)-space represents the similarity of two artifacts with respect to the concepts they share. Thus, terms that occur in similar artifacts will be near each other in the \( k \)-space of concepts, even if they never co-occur in the same artifact. On the other hand, some artifacts that do not share any word, but share similar words may nonetheless be near in the \( k \)-space.

D. Latent Dirichlet Allocation

LDA is a generative probabilistic model where documents are modeled as random mixtures over latent topics [12]. Recent applications of LDA for software maintenance and evolution task include identifying cross-cutting concerns [18], extracting domain topics from source code [19], automatic categorization of software systems [20], windowed topic analysis [21], and measuring class cohesion [13]. 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 \( N \sim \text{Poisson distribution} (\xi) \)
2) Choose \( \theta \sim \text{Dirichlet distribution} (\alpha) \)
3) For each of the \( N \) words \( w_n \):
   a) Choose a topic \( t_n \sim \text{Multinomial} (\theta) \).
   b) Choose a word \( w_n \) from \( p(w_n|t_n, \beta) \), a multinomial probability conditioned on topic \( t_n \).

To compare document similarities we use Hellinger Distance, a symmetric similarity measure between two probability distributions. Prior research in topic modeling [22] have also applied this as a means of capturing similarity between documents. Hellinger Distance is defined as:

\[
H(P, Q) = \sum_{x \in X} \left( \sqrt{P(x)} - \sqrt{Q(x)} \right)^2
\]
where $P$ and $Q$ are probability distributions, $X$ is the set of possible discrete random variables and $P(x)$ and $Q(x)$ correspond to probability of the discrete random variable $x$ occurring for the distributions $P$ and $Q$ respectively.

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, in our study we used different values of $T$ aiming at analyzing its impact on the recovery accuracy.

IV. CASE STUDY

This section describes in detail the design of a case study carried out to assess the equivalence of the exploited IR methods and the accuracy of the novel LDA-based traceability recovery technique. The case study was conducted following the guidelines given by Yin [23].

A. Definition and Context

The goals of the case study were (i) analyzing the recovery accuracy provided by the exploited IR methods; (ii) analyzing whether or not different types of IR-based traceability recovery methods provide orthogonal similarity measures between software artefacts; and (iii) analyzing whether or not the combination of different IR methods improves the traceability recovery accuracy with respect to stand-alone methods. Thus, the quality focus of our case study was on ensuring better recovery accuracy, while the perspective was both of (i) a researcher, who wants to compare the accuracy of different IR methods for traceability recovery; and of (ii) a project manager, that wants to evaluate the possibility of adopting a particular IR method or a combination of different IR methods for traceability recovery.

The case study was conducted on two software repositories, i.e., EasyClinic and eTour. The former is a software system providing support to manage a medical doctor’s office, while the latter is an electronic tourist guide. Table I shows the characteristics of the considered software systems. The table shows the size of the system in terms of lines of code (LOC), the number of use cases (UCs) as well as the number of source code classes (CCs). The table also reports the number of possible and correct links between use cases and classes, respectively. The number of possible links is given by the product of the number of use cases with the number of classes. The traceability information 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. The term-by-document matrices and the oracles of both the systems are available for replication purposes.

B. Research Questions and Planning

In the context of our case study we formulated three research questions (RQ):

1. **RQ1**: Which is the IR method that provides the more accurate list of candidate links?
2. **RQ2**: Do different types of IR methods provide orthogonal similarity measures?
3. **RQ3**: Does the combination of different IR methods overcome the accuracy of stand-alone IR methods?

To address the above research questions, the studied IR methods were used to recover traceability links between the use cases and the code classes of EasyClinic and eTour.

The case study was organized in two phases. In the first phase we compared the accuracy as well as the similarity measures provided by different IR-based traceability recovery methods. Each IR method is provided identical term-by-document matrices as input. In order to cover a large number of IR methods, we selected the JS method, VSM, LSI and LDA. The first three methods were previously used for traceability recovery [3], [4], [5], [6], [7], [8], [9], [10], [11], while LDA was never used for such a task.

In the second phase of the study, we combined different IR methods according to the results achieved in the first phase. Then, we compared the accuracy of the combined IR methods with the accuracy achieved with stand-alone methods, i.e., VSM, JS, LSI, and LDA. To combine two IR techniques we apply affine transformation [24], a technique which was previously used to combine results of the feature location techniques [25]. The transformation first maps similarity results to a standard normal distribution as follows:

$$sim_{m_1}(x, y) = \frac{m_i(x, y) - mean(m_i(X, Y))}{stdev(m_i(X, Y))}$$

where $X, Y$ are sets of software artifacts, $x \in X, y \in Y$ and $sim_{m_1}(x, y)$ 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 $m_i$. Then, the affine transformation combining the two techniques is defined as:

$$sim_{combined}(x, y) = \lambda sim_{m_1}(x, y) + (1 - \lambda) sim_{m_2}(x, y)$$

where $\lambda \in [0, 1]$ expresses confidence in each technique.

C. Data Collection and Analysis

For each tracing activity the number of correct links and false positives retrieved by each exploited IR method were collected aiming at evaluating its accuracy. A tool simulating the behavior of the software engineer during the classification of the candidate links automatically identified

<table>
<thead>
<tr>
<th>LOC</th>
<th>UCs</th>
<th>CCs</th>
<th>Correct links</th>
<th>Possible links</th>
</tr>
</thead>
<tbody>
<tr>
<td>EasyClinic</td>
<td>15,000</td>
<td>50</td>
<td>47</td>
<td>93</td>
</tr>
<tr>
<td>eTour</td>
<td>45,000</td>
<td>58</td>
<td>116</td>
<td>366</td>
</tr>
</tbody>
</table>

Table I

CHARACTERISTICS OF THE SOFTWARE SYSTEMS.

---

2http://www.cs.wm.edu/semereu/data/icpc10-tr-lda.
the number of correct links and false positives. The tool takes as an input the ranked list of candidate links built by the exploited IR method and classifies each link as correct or false positive by exploiting the original traceability matrix.

For the comparison of different IR methods and their combination (related to questions RQ1 and RQ2), we used two well-known Information Retrieval (IR) metrics, namely recall and precision [1]:

\[
\text{recall} = \frac{|\text{correct} \cap \text{retrieved}|}{|\text{correct}|} \%
\]

\[
\text{precision} = \frac{|\text{correct} \cap \text{retrieved}|}{|\text{retrieved}|} \%
\]

where \(\text{correct}\) and \(\text{retrieved}\) represent the set of correct links and the set of links retrieved by the tool, respectively.

To identify whether different types of IR methods provide orthogonal similarity measures – i.e., RQ2 – we statistically analyzed the similarity measures provided by the selected IR methods. Such an analysis uses PCA, a statistical technique capable of identifying the various orthogonal dimensions captured by the data (principal components) and which measure contribute 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 insight on the orthogonality between similarity metrics. Moreover, to have a further comparison of the traceability retrieval methods we used the following overlap metrics:

\[
\text{correct}_{m_i \cap m_j} = \frac{|\text{correct}_{m_i} \cap \text{correct}_{m_j}|}{|\text{correct}_{m_i} \cup \text{correct}_{m_j}|} \%
\]

\[
\text{correct}_{m_i \setminus m_j} = \frac{|\text{correct}_{m_i} \setminus \text{correct}_{m_j}|}{|\text{correct}_{m_i} \cup \text{correct}_{m_j}|} \%
\]

where \(\text{correct}_{m_i}\) represents the set of correct links identified by the IR method \(m_i\). It is worth noting that \(\text{correct}_{m_i \cap m_j}\) measures the overlap between the set of correct links retrieved by the two IR methods, while \(\text{correct}_{m_i \setminus m_j}\) 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 to enriching the set of correct links identified by the other method.

**D. Threats to validity**

Concerning the generalization of the results (i.e., external validity), an important threat is related to the repositories used in the study. EasyClinic is not comparable to industrial projects, but repositories used by other authors to compare different IR methods have a comparable size [4], [10], [8]. Moreover, EasyClinic was previously used to evaluate IR methods to recover traceability links [6]. To mitigate such a threat we replicated the case study on eTour, a larger software system. To the best of our knowledge eTour is one of the largest repository used for studying IR methods in the context of traceability link recovery.

Regarding the relationship between theory and observation (i.e., construct validity), recall and precision are widely used metrics for assessing an IR technique. Moreover, the overlap metrics give a good indication on the overlap of the correct links recovered by the different IR methods. Finally, the similarity measures provided by each IR method are also statistically analyzed using PCA aiming at verifying the presence of IR methods that provide orthogonal similarity measures.

The accuracy of the oracle used to evaluate the studied traceability recovery methods could also affect the achieved results. To mitigate such a threat we used the original traceability matrices provided by the original developers. Moreover, the links contained in these matrices were validated during review meetings made by the original development team together with PhD students and academic researchers.

**V. EXPERIMENTAL RESULTS**

In this section we analyze and discuss the results achieved and provide answers to our research questions.

**A. Accuracy of the Experimented IR Methods**

Table II provides precision and recall for the exploited techniques, on both EasyClinic and eTour, when various fixed cut points are applied. Precision and recall are computed using the top \(\mu\) candidate traceability links for each fixed cut point. The results indicate that none of the standalone techniques outperforms all the others, but three of the techniques (JS, LSI, and VSM) display virtually identical performances. For all investigated cut points, one of the three techniques, JS, LSI, or VSM, boasts the highest accuracy. Accuracy of the remaining IR-based technique (i.e., LDA) fails in comparison to the three top performing methods. Those remaining rows in Table II correspond to various configurations of LDA where each configuration differs in the number of topics derived when LDA was applied on the respective corpus. We varied the number of topics starting at 50 and incrementing by 50 until we considered 300 topics, as prior work has shown its best to repeat the topic extraction process until an appropriate number of topics is determined [19]. We varied the number of topics to obtain insight on its impact on traceability recovery accuracy. Although no configuration of LDA provided accuracy comparable to the top three techniques, we identify the configuration with 250 topics as the best across both systems. From this point forward the configuration with 250 topics will represent the LDA-based traceability recovery technique in our analysis.

**B. Equivalence of the Experimented IR Methods**

The second step of our analysis aims at verifying whether different IR-based techniques are able to capture orthogonal information. Orthogonality of IR-based techniques may
We evaluate the overlap of the set of correct links of the top candidate links for specific cut points. Given two techniques, recovery methods, focusing on the overlap between sets of then also utilizing the technique based on LDA.

In this dimension, our findings indicate that capturing the variable, LDA, indicating that it is the only major factor VSM. PC

1 variables highly correlated to PC

2 of the variance of the data for EasyClinic and eTour. The respective, while PC

73.79% of the variance in the data for EasyClinic and eTour.

By analyzing the percentage of correct links that overlap between techniques we can identify orthogonality with regards to correct links omitted by other IR-based techniques. As an initial step toward investigating this we perform PCA using the similarity values of all candidate links for each technique. For the two systems evaluated the results (see Table III) indicate that two principal components account for a significant percentage of the variation in the data. PC

1 accounts for 76.15% and 73.79% of the variance in the data for EasyClinic and eTour respectively, while PC

2 accounts for 23.64% and 25.11% of the variance of the data for EasyClinic and eTour. The variables highly correlated to PC

1 include JS, LSI, and VSM. PC

2, on the other hand, has only one highly correlated variable, LDA, indicating that it is the only major factor in this dimension. Our findings indicate that capturing the two significant dimensions in the data requires utilizing one technique from the set containing JS, LSI, and VSM and then also utilizing the technique based on LDA.

We further examine the orthogonality of traceability recovery methods, focusing on the overlap between sets of candidate links for specific cut points. Given two techniques, we evaluate the overlap of the set of correct links of the top \( \mu \) candidate links, where \( \mu \) is the cut point. The information gleaned from evaluating overlap is more specific to the actual application of the techniques. More specifically, through analyzing the percentage of correct links that overlap between techniques we can identify orthogonality with regards to correct links identified. For example, if two techniques consistently return sets of correct links which have little overlap those techniques may be orthogonal. Each technique is providing insight complementary to the other. Therefore, through evaluating overlap we can determine whether a technique provides different correct links or whether it provides only a subset of the correct links returned by another technique. Based on the results of PCA we decided to consider only combinations, which include the LDA-based traceability recovery technique. Table IV contains results showing the percentage of overlap between various combinations of IR techniques for both EasyClinic and eTour. The percentages represent the portion of correct links identified by the LDA-based method which also appear in the set of correct links identified by the other method. For example, if the LDA-based method returned three correct links and the JS-based method returned eight correct links, and one of the three links identified by the LDA-based technique is also identified by the JS-based technique, there is an overlap of 10% of the correct links (\( 1/ (3 + 8 - 1) \cdot 100 = 10\% \)). For both systems overlap between the candidate sets is relatively low. This indicates that in those two cases sets of candidate links have few links in common. Actually, this result is quite expected because of (i) the lower accuracy of LDA as compared to the other methods and (ii) the results of the PCA. Nevertheless, the results in Table IV indicate that in many cases LDA-based method is capable of identifying correct links, which are not obtained in the results by other IR techniques, especially in the case of eTour software. For eTour the percentage of correct links found using the LDA-based method and missed using another technique is about 10%. The results for EasyClinic, on the other hand are not so encouraging. This is, in part, because of the superb accuracy obtained by the canonical techniques, i.e., JS, VSM, and LSI. Their performance limits the number of correct links possible for LDA-based technique to uniquely identify in this case. But overall across both systems the potential insight that LDA-based traceability recovery method may provide appear promising. Minimal overlap presents the possibility of augmenting techniques and obtaining \( \mu \) candidate links with accuracy superior to canonical techniques. Our results indicate that LDA-based technique’s candidate links contain correct links omitted by other IR-based techniques.

C. Combination of LDA with other IR methods

Analysis of the overlap between the LDA-based technique and other techniques reveals the opportunity to augment techniques to improve accuracy. The analysis of the overlap between the correct links identified by the two techniques

<table>
<thead>
<tr>
<th>EasyClinic</th>
<th>js</th>
<th>lda(250)</th>
<th>lda(200)</th>
<th>lda(150)</th>
<th>lda(100)</th>
<th>lda(50)</th>
<th>vsm</th>
<th>lsi</th>
<th>js</th>
<th>lda(250)</th>
<th>lda(200)</th>
<th>lda(150)</th>
<th>lda(100)</th>
<th>lda(50)</th>
<th>vsm</th>
<th>lsi</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.095</td>
<td>0.087</td>
<td>0.056</td>
<td>-0.017</td>
<td>0.993</td>
<td>0.041</td>
<td>-0.101</td>
<td>-0.047</td>
<td>0.000</td>
<td>-0.001</td>
<td>-0.092</td>
<td>0.996</td>
<td>0.017</td>
<td>-0.004</td>
<td>0.000</td>
<td>-0.003</td>
</tr>
</tbody>
</table>
highlight LDA's ability to identify correct links unique to its set. Here we investigate the impact of combining LDA with other techniques on the accuracy of candidate links returned. We consider only combinations which include LDA, in part because of the results obtained in Section V-B.

The results of combining LDA with other IR-based techniques appear in Table V. Note that we evaluate lambda values from 0.00 to 1.00 with increments of 0.05. Because of space limitations in Table V we report a subset of the complete results, i.e., the results providing the best overall accuracy. Results indicate that in many cases combining LDA with canonical methods improves accuracy of candidate links. Values bolded in the table represent the highest accuracy its performance is still comparable to that of the canonical technique. For example, results for combining LDA with LSI for eTour when lambda is 0.10 are comparable to the stand-alone version of LSI for the cut points evaluated.

Analysis of Lambda: Figure 2 demonstrates the impact varying lambda between zero and one have on the F-Measure. The F-measure is a weighted harmonic mean of precision and recall and calculated as:

\[
F\text{-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

which serves as an indicator of combined precision and recall values. Note that in the two cases, where lambda is 0 and lambda is 1, ranking is based solely on the canonical technique and LDA-based technique respectively. Increasing lambda amplifies LDA's influence on the resulting ranked list. Figure 2 shows that initially increasing the influence of the LDA-based technique slightly enhances accuracy. This trend continues until lambda reaches 0.20. From that point forward, as lambda continues to approach 1, accuracy decreases drastically. Once lambda exceeds 0.85 in the example shown in Figure 2 all the combinations display virtually identical results for F-Measure. This can be attributed to LDA's dominate influence due to the large value of lambda. Overall, adding more weight to LDA at least allows accuracy to remain stable and in some cases improve until a certain point is reached. In both cases assigning approximately 10% – 25% of the weight to the LDA-based technique results in the optimal performance.

VI. LESSONS LEARNED AND OPEN ISSUES

The results achieved in the reported case study provided us with a number of lessons learned (LL) and open issues (OI):

- **LL1**: JS, VSM, LSI are able to provide almost the same information when used for documentation-to-
techniques (i.e., JS or VSM) should be preferred to the more complex method (i.e., LSI). In particular, the cost of computing SVD for LSI may not be compensated by an improvement in the retrieval accuracy.

- **LL7**: LDA is able to capture some information missed by VSM, LSI, and JS when used for recovering traceability links between code and documentation. The second principal component highlighted by the PCA has only one highly correlated variable, LDA, indicating that it is the only major factor of the dimension. This means that LDA is able to capture a dimension unique to the set of techniques which we considered. The results achieved with the PCA are also confirmed by the analysis of the overlap between sets of candidate links. In particular, the results achieved indicate that in many cases LDA-based method is capable of identifying correct links not included in the results of other techniques, especially in the case of eTour.

- **LL8**: LDA is not able to provide more accurate lists of candidate links when compared to JS, VSM, and LSI. To evaluate the accuracy of the LDA-based method we vary the number of topics starting at 50 and incrementing by 50 until we consider 300 topics. The best results were achieved when the number of topics is 250. The analysis of precision and recall demonstrates that the LDA-based technique results in lower accuracy than the other three techniques. In particular, at the same recall level the precision achieved by JS, VSM, and LSI are generally twice better than the precision level achieved by LDA.

- **OO1**: The combination of LDA with other IR methods should be further explored aiming at sensibly improving the accuracy of the stand-alone methods. Although the accuracy of LDA is somewhat low, it is able to identify correct links that are missed by canonical methods, i.e., JS, VSM, and LSI. For this reason, we propose to combine LDA with other techniques using affine transformation. Even though the results achieved seem promising, the combined method does not sensibly outperform the stand-alone methods. Indeed, the advantages of the combination should be much more evident to justify the overall computation costs of the combined method. Identifying better combination strategy poses another research question that we intend to further investigate in our future work.

### VII. Conclusion and Future Work

In this paper we presented a case study aiming at evaluating the equivalence of different IR methods (i.e., JS, VSM, LSI, and LDA) when used for traceability recovery. The exploited IR methods were used to recover traceability links between use cases and code classes of two software repositories. The results for LDA-based traceability recovery technique indicated lower performance as compared to other

code traceability recovery. The results of PCA show two dominate principal components for each software system. These two components account for about 75% and 25% of the variance of the data for both systems. The variables highly correlated to PC1 include JS, LSI, and VSM. Such results suggest that these IR methods are able to capture the majority of the information contained in the artifact content but they provide almost the same information. Such a result suggests that the simpler
IR-based techniques. However, while JS, VSM, LSI are equivalent, LDA is able to capture a dimension unique to the set of techniques which we considered. Such results suggest that LDA is able to provide orthogonal similarity metrics as compared to the other exploited IR methods and it can be used to augment stand-alone methods and obtain improved accuracy. Based on these results we propose a method to combine LDA with VSM, LSI, and JS aiming at improving the accuracy of the stand-alone methods. The results achieved seem promising but the combination of LDA with other IR methods should be further explored aiming at sensibly improving the accuracy of the stand-alone methods.

Future work will be devoted to further study the proposed combined method and assess the equivalence of different IR methods when used for traceability link recovery. Moreover, there are a number of directions on how to improve the accuracy of the proposed traceability recovery method. A first direction aims at defining a more sophisticated method for combining LDA with other IR methods, including utilizing structural information [26]. A second direction aims at integrating a specialized learning algorithm exploiting the relevance feedback analysis into the approach.

REFERENCES


