# 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
roliveto@unisa.it, mgethers@cs.wm.edu, denys@cs.wm.edu, adelucia@unisa.it

***Abstract*—We present an empirical study to statistically analyze the equivalence of several traceability recovery methods based on Information Retrieval (IR) techniques. 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 the Jensen-Shannon (JS) method, Vector Space Model (VSM), Latent Semantic Indexing (LSI), and Latent Dirichlet Allocation (LDA). The results show that while JS, VSM, and LSI are almost equivalent, LDA is able to capture a dimension unique to the set of techniques which we considered.**

***Keywords*-Traceability Recovery; Information Retrieval; Empirical Studies.**

## I. Introduction

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 (e.g., [3], [4]). 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 Model (VSM), probabilistic model [1], and 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 the experimented 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 [5], VSM, LSI, and Latent Dirichlet Allocation (LDA) [6]. The first three methods were selected because they are widely used and seem to be the techniques that give the best results [3], [4], [5]. LDA is not as widely used for traceability link recovery though it has been used recently [7]. However, we also experiment such a technique for traceability recovery because LDA is able to capture some aspects missed by other IR methods, such as LSI, when it is used in other contexts [8].

The empirical analysis has been conducted on two software repositories, namely EasyClinic and 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 prove that the accuracy of LDA is lower than previously used methods. However, while JS, VSM, and LSI are almost equivalent, LDA is able to capture some information missed by the other exploited IR methods. These considerations suggest that probably LDA can be used as a method to augment canonical methods—e.g., JS, VSM, and LSI—aiming at improving their accuracy.

The rest of the paper is organized as follows. Section II discusses related work, while Section III briefly describe an IR-based traceability recovery process. Sections IV and V provide details on the design of the case study and report the results achieved, respectively. Section VI gives concluding remarks.

## II. Related Work

The use of IR methods for traceability recovery was introduced by Antoniol *et al.* [3]. They apply VSM and the probabilistic model [1] to trace source code onto software documentation. Later, other IR methods (e.g., LSI, JS method and Numerical Analysis) have been proposed to recover links between different types of artifacts [4], [5], [9]. In particular, IR methods have been used to recover traceability between requirements [10], between requirements and design artifacts [11], between maintenance requests and software documents [12], and between other types of artifacts [13], [14].

All these reported 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 carried out to analyze the equivalence of different IR-based traceability recovery approaches.

## III. IR-based Traceability Recovery

IR methods index documents and query in a document space 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 term extraction is preceded by a text normalization phase. In particular, in our study we pruned out white spaces and most non-textual tokens (e.g., special symbols, numbers) from the artifact contents. We also used a stop word list to discard common terms (e.g., articles, adverbs) that are not useful to characterize the semantics of the artifact [1]. We also performed a morphological analysis, i.e., stemming [15], on the extracted terms to remove suffixes of words to extract their stems.

The extracted information is 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 $a_{i,j}$ 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 *tf-idf* schema [1].

Based on the *term-by-document matrix* representation, different IR methods can be used to rank the similarity between the pairs of artifacts. Then a threshold (e.g., a cut point [3]) is used to select the first $\mu$ documents in the ranked list. Thus, any IR method will fail to retrieve some of the correct links, as well as retrieve links between artifacts that are not correct (false positives).

## IV. Case Study

This section reports the design of our case study that was conducted following the guidelines given by Yin [16].

### A. Definition and Context

The *goals* of the case study were analyzing the recovery accuracy provided by the different IR methods and analyzing whether or not different types of IR-based traceability recovery methods provide orthogonal similarity measures between software artifacts

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 touristic 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*), and the number of source code classes (*CCs*). The table also reports the number of correct links between use cases and 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[1].

Table I
Characteristics of the software systems.

| | LOC | UCs | CCs | Correct links |
|---|---|---|---|---|
| EasyClinic | 15,000 | 30 | 47 | 93 |
| eTour | 45,000 | 58 | 116 | 366 |

### B. Research Questions and Planning

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

- **RQ**$_1$: Which is the IR method that provides the more accurate list of candidate links?
- **RQ**$_2$: Do different types of IR methods provide orthogonal similarity measures?

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. 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 (e.g., [3], [4], [5]). LDA has been also recently used for traceability link recovery [7].

### C. Data Collection and Analysis

To evaluate the accuracy of the experimented techniques we collected the number of correct links and false positives retrieved by each exploited IR method. We used a tool that simulate the behavior of the software engineer during the classification of the candidate links. 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 (**RQ**$_1$) we used two well-known Information Retrieval (IR) metrics, namely recall and precision [1]. Moreover, to identify whether different types of IR methods provide orthogonal similarity measures (**RQ**$_2$) 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

[1] http://www.cs.wm.edu/semeru/data/icpc10-tr-lda.

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 main contributors to those components. This information provides insight on the orthogonality between similarity metrics. Also, to have a further comparison of the traceability retrieval methods we used the following overlap metrics:

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

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

where $correct_{m_i}$ represents the set of correct links identified by the IR method $m_i$. It is worth noting that $correct_{m_i \cap m_j}$ measures the overlap between the set of correct links retrieved by the 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 to enriching the set of correct links identified by the other method.

### D. Threats to validity

An important threat is related to the repositories used in the study. They are not comparable to industrial projects, but repositories used by other authors to compare different IR-based traceability recovery methods have a comparable size [3], [4], [10]. Moreover, EasyClinic was used as object systems in the traceability recovery challenge organized at TEFSE 2009[2]. To the best of our knowledge the two systems are among the largest repositories used for studying IR methods in the context of traceability link recovery.

The accuracy of the experimented methods has been evaluated using recall and precision, two metrics widely used for assessing an IR technique. PCA and the overlap metrics give a good indication on the orthogonality of the similarity measures provided by the different IR methods.

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.

## V. Experimental Results

In this section we analyze and discuss the results achieved and provide answers to our research questions. More details can be found in our technical report[3].

### 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. 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 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. The results of PCA show two principal components (PC) 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. Thus, to capture the two significant dimensions in the data it is needed to use LDA and one technique from the set containing JS, LSI, and VSM.

We also analyze 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 allows us to 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 technique. Table III contains results showing the percentage of overlap between various combinations of IR techniques for eTour (see our technical report for complete results). 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

[2]http://web.soccerlab.polymtl.ca/tefse09/Challenge.htm

[3]http://www.cs.wm.edu/semeru/papers/IR01.pdf.

Table II
RESULTS OF PRECISION (PR) AND RECALL (R) FOR VARIOUS CUT POINTS OF BOTH EASYCLINIC AND ETOUR.

| | EasyClinic (the total number of correct links is 93) | | | | | | | | | | | | | | | | eTour (the total number of correct links is 366) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 | | 10 | | 25 | | 50 | | 100 | | 200 | | 300 | | 500 | | 25 | | 50 | | 75 | | 100 | | 300 | | 500 | | 700 | | 1000 | |
| | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R | Pr | R |
| js | 80 | 4 | 60 | 6 | 52 | 14 | 50 | 27 | 45 | 48 | 32 | 68 | 26 | 83 | 17 | 91 | 76 | 5 | 72 | 10 | 60 | 12 | 56 | 15 | 34 | 28 | 26 | 36 | 22 | 42 | 17 | 47 |
| lsi | 60 | 3 | 50 | 5 | 56 | 15 | 52 | 28 | 46 | 49 | 32 | 68 | 26 | 84 | 17 | 90 | 64 | 4 | 50 | 7 | 44 | 9 | 40 | 11 | 28 | 23 | 25 | 34 | 21 | 40 | 17 | 46 |
| vsm | 80 | 4 | 60 | 6 | 52 | 14 | 50 | 27 | 43 | 46 | 32 | 68 | 25 | 81 | 17 | 90 | 68 | 5 | 70 | 10 | 60 | 12 | 53 | 15 | 33 | 27 | 27 | 37 | 21 | 41 | 17 | 47 |
| lda(50) | 40 | 2 | 20 | 2 | 12 | 3 | 8 | 4 | 9 | 10 | 9 | 19 | 9 | 30 | 9 | 49 | 8 | 1 | 6 | 1 | 5 | 1 | 5 | 1 | 6 | 5 | 6 | 9 | 7 | 13 | 7 | 18 |
| lda(100) | 20 | 1 | 10 | 1 | 8 | 2 | 12 | 6 | 10 | 11 | 9 | 19 | 8 | 27 | 9 | 51 | 4 | 1 | 6 | 1 | 4 | 1 | 5 | 1 | 8 | 7 | 8 | 11 | 8 | 15 | 7 | 20 |
| lda(150) | 20 | 1 | 10 | 1 | 16 | 4 | 12 | 6 | 12 | 13 | 10 | 20 | 9 | 29 | 10 | 56 | 8 | 1 | 4 | 1 | 4 | 1 | 5 | 1 | 6 | 5 | 7 | 10 | 7 | 14 | 7 | 19 |
| lda(200) | 20 | 1 | 30 | 3 | 16 | 4 | 10 | 5 | 9 | 10 | 7 | 15 | 9 | 29 | 11 | 60 | 4 | 1 | 16 | 2 | 13 | 3 | 11 | 3 | 9 | 7 | 6 | 9 | 6 | 12 | 6 | 17 |
| lda(250) | 20 | 1 | 30 | 3 | 12 | 3 | 10 | 5 | 12 | 13 | 9 | 19 | 11 | 34 | 11 | 59 | 8 | 1 | 10 | 1 | 7 | 1 | 7 | 2 | 5 | 4 | 7 | 10 | 7 | 14 | 7 | 18 |
| lda(300) | 0 | 0 | 20 | 2 | 12 | 3 | 12 | 6 | 10 | 11 | 12 | 25 | 10 | 31 | 10 | 54 | 16 | 1 | 10 | 1 | 9 | 2 | 8 | 2 | 9 | 8 | 8 | 11 | 8 | 15 | 7 | 18 |

Table III
ETOUR: OVERLAP OF CANDIDATE LINKS OF LDA-BASED TECHNIQUE AND OTHER IR-BASED TECHNIQUES.

| | eTour | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 25 | 50 | 75 | 100 | 300 | 500 | 700 | 1000 |
| $correct_{LDA \cap JS}$ | 0% | 5% | 4% | 5% | 9% | 19% | 25% | 27% |
| $correct_{LDA \setminus JS}$ | 10% | 8% | 6% | 7% | 6% | 6% | 6% | 8% |
| $correct_{LDA \cap VSM}$ | 0% | 5% | 4% | 5% | 10% | 17% | 25% | 26% |
| $correct_{LDA \setminus VSM}$ | 11% | 8% | 6% | 7% | 6% | 8% | 7% | 9% |
| $correct_{LDA \cap LSI}$ | 13% | 11% | 9% | 9% | 15% | 22% | 28% | 30% |
| $correct_{LDA \setminus LSI}$ | 0% | 7% | 6% | 7% | 3% | 5% | 5% | 7% |

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 III 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. 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 show that LDA-based technique's candidate links contain correct links omitted by other IR-based techniques.

## VI. CONCLUSION AND FUTURE WORK

We reported a case study to evaluate the equivalence of different IR methods (i.e., JS, VSM, LSI, and LDA) when used for traceability recovery. The results achieved demonstrated that the LDA-based traceability recovery technique provided lower accuracy as compared to other 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.

Future work will be devoted to further assess the equivalence of different IR methods when used for traceability link recovery. Moreover, we also plan to combine canonical IR methods with LDA in order to improve the accuracy of stand-alone methods.

## REFERENCES


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

[2] 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.

[3] 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, pp. 970–983, 2002.

[4] A. Marcus and J. I. Maletic, "Recovering documentation-to-source-code traceability links using latent semantic indexing," in *Proc. of ICSE '03*, 2003, pp. 125–135.

[5] A. Abadi, M. Nisenson, and Y. Simionovici, "A traceability technique for specifications," in *Proc. of IEEE ICPC '08*, 2008, pp. 103–112.

[6] 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.

[7] H. Asuncion, A. Asuncion, and R. Taylor, "Software Traceability with Topic Modeling," *IEEE ICSE'10*, 2010.

[8] Y. Liu, D. Poshyvanyk, R. Ferenc, T. Gyimóthy, and N. Chrisochoides, "Modeling class cohesion as mixtures of latent topics," in *Proc. of IEEE ICSM '09*, 2009, pp. 233–242.

[9] G. Capobianco, A. De Lucia, R. Oliveto, A. Panichella, and S. Panichella, "Traceability recovery using numerical analysis," in *Proc. of WCRE '09*, 2009, pp. 195–204.

[10] 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.

[11] J. Cleland-Huang, R. Settimi, C. Duan, and X. Zou, "Utilizing supporting evidence to improve dynamic requirements traceability," in *Proc. of IEEE RE '05*, 2005, pp. 135–144.

[12] G. Antoniol, G. Canfora, G. Casazza, and A. De Lucia, "Identifying the starting impact set of a maintenance request," in *Proc. of 4th CSMR '00*, 2000, pp. 227–230.

[13] 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.

[14] M. Lormans, A. van Deursen, and H.-G. Groß, "An industrial case study in reconstructing requirements views," *Empirical Software Engineering*, vol. 13, no. 6, pp. 727–760, 2008.

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

[16] R. K. Yin, *Case Study Research: Design and Methods*, 3rd ed. SAGE Publications, 2003.