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

Rocco Oliveto, Malcom Gethers, Denys Poshyvanyk, Andrea De Lucia
Traceability Management

• Traceability…
  - “the ability to describe and follow the life of an artifact, in both a forwards and backwards direction”

• Maintaining traceability between software artifacts is important for software development and maintenance
  • program comprehension
  • impact analysis
  • software reuse
Traceability Link Recovery

- Most software artifacts contain text.
- Conjecture: artifacts having a high text similarity are likely good candidates to be traced onto each other.
- IR techniques can be used to calculate the similarity between software artifacts.
Tracing Software Artifacts Using IR Methods

Source Artifacts → Text pre-processing → Word extraction and filtering → Indexing
- Artifact corpus - → Classifier
- Artifact to artifact ranking - → Cutter
- Top part of the ranking - → Classifies

Source Artifacts → Text pre-processing → Word extraction and filtering → Classifier
- Artifact corpus - → Cutter
- Top part of the ranking - → Classifies
Classifier: two basic models

- **Probabilistic model**
  - The similarity between a source and a target artifact is based on the probability that the target artifact is related to the source artifact (i.e., Jensen-Shannon)

- **Vector space model**
  - Source and target artifacts are represented in a vector space (of terms) and the similarity is computed through vector operations

- **Improvements to basic models:**
  - Latent Semantic Indexing
  - Latent Dirichlet Allocation
Vector Space Model

- Software artifacts are represented as vectors in the space of terms (vocabulary)
- Vector values might be values (the term is or is not in the artifact)
- Usually computed as the product of a local and a global weights
  - Local weight: based on the frequency of occurrences of the term in the document
  - Global weight: the more the term is spread in the artifact space the less it is relevant to the subject document
Latent Semantic Indexing

• Extension of the Vector Space Model based on Singular Value Decomposition (SVD)
  • The term-by-document matrix is decomposed into a set of k orthogonal factors from which the original matrix can be approximated by linear combination

• Overcomes some of the deficiencies of assuming independence of words (co-occurrences analysis)
  • Provides a way to automatically deal with synonymy
  • Avoids preliminary text pre-processing and morphological analysis (stemming)
Latent Dirichlet Allocation

• LDA is a generative probabilistic model where documents are modeled as random mixtures over latent topics.

• LDA is similar to pLSA, except that in LDA the topic distribution is assumed to have a Dirichlet distribution.

• We use Hellinger distance, a symmetric similarity measure between two probability distributions.
Motivation

• No empirical studies on evaluating multiple IR methods for traceability link recovery:
  - Latent Semantic Indexing (LSI)
  - Vector Space Model (VSM)
  - Jenson-Shannon (JS)
  - Latent Dirichlet Allocation (LDA)

• Some studies indicate controversial results

• Which IR technique should I use?
Empirical Assessment of Traceability Link Recovery Techniques

• Research questions (RQ)
  • RQ1: Which is the IR method that provides the more accurate list of candidate links?
  • RQ2: Do different types of IR methods provide orthogonal similarity measures?

• Design of the case studies
  • EasyClinic and eTour software systems
    • EasyClinic: 93 out of 1,410 possible links
    • eTour: 364 out of 6,728 possible links
  • IR techniques: JS, VSM, LSI and LDA
  • Case study data: [link](www.cs.wm.edu/semeru/data/icpc10-tr-lda)
RQ₁ - Traceability Link Recovery Accuracy
RQ₁ - Traceability Link Recovery Accuracy

EasyClinic

Graph showing the recall at different cut points for JS, LDA, LSI, and VSM methods.
RQ₁ - Traceability Link Recovery Accuracy
RQ\textsubscript{1} - Traceability Link Recovery Accuracy
RQ₂ - Principal Component Analysis (PCA)

- Do different types of IR methods provide orthogonal similarity measures?

- PCA procedure:
  - collect data
  - identify outliers
  - perform PCA
# PCA Results: Rotated Components

<table>
<thead>
<tr>
<th></th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion</td>
<td>73.79</td>
<td>25.11</td>
<td>0.96</td>
<td>0.14</td>
</tr>
<tr>
<td>Cumulative</td>
<td>73.79</td>
<td>98.9</td>
<td>99.86</td>
<td>100</td>
</tr>
<tr>
<td>JS</td>
<td>0.993</td>
<td>0.041</td>
<td>-0.101</td>
<td>-0.047</td>
</tr>
<tr>
<td>LDA(250)</td>
<td>-0.092</td>
<td>0.996</td>
<td>0.017</td>
<td>-0.004</td>
</tr>
<tr>
<td>LSI</td>
<td>0.986</td>
<td>-0.046</td>
<td>0.158</td>
<td>-0.01</td>
</tr>
<tr>
<td>VSM</td>
<td>0.992</td>
<td>0.097</td>
<td>-0.055</td>
<td>0.057</td>
</tr>
</tbody>
</table>
RQ$_2$ - Overlap Among Techniques

- Do different types of IR methods provide orthogonal similarity measures?
- Overlap Metrics

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

\[
\text{correct}_{m_i \setminus m_j} = \frac{\text{correct}_{m_i \setminus m_j}}{\text{correct}_{m_i \cup m_j}} \%
\]
### Results for Overlap Metrics for eTour

<table>
<thead>
<tr>
<th></th>
<th>25</th>
<th>50</th>
<th>75</th>
<th>100</th>
<th>300</th>
<th>500</th>
<th>700</th>
<th>1K</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct LDA\JS</td>
<td>0%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>9%</td>
<td>19%</td>
<td>25%</td>
<td>27%</td>
</tr>
<tr>
<td>correct LDAnJS</td>
<td>10%</td>
<td>8%</td>
<td>6%</td>
<td>7%</td>
<td>6%</td>
<td>6%</td>
<td>6%</td>
<td>8%</td>
</tr>
<tr>
<td>correct LDA\VSM</td>
<td>0%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>10%</td>
<td>17%</td>
<td>25%</td>
<td>26%</td>
</tr>
<tr>
<td>correct LDAnVSM</td>
<td>11%</td>
<td>8%</td>
<td>6%</td>
<td>7%</td>
<td>6%</td>
<td>8%</td>
<td>7%</td>
<td>9%</td>
</tr>
<tr>
<td>correct LDA\LSI</td>
<td>13%</td>
<td>11%</td>
<td>9%</td>
<td>9%</td>
<td>15%</td>
<td>22%</td>
<td>28%</td>
<td>30%</td>
</tr>
<tr>
<td>correct LDAnLSI</td>
<td>0%</td>
<td>7%</td>
<td>6%</td>
<td>7%</td>
<td>3%</td>
<td>5%</td>
<td>5%</td>
<td>7%</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>50</td>
<td>75</td>
<td>100</td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>1K</td>
</tr>
<tr>
<td>----------------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>correct LDA\JS</td>
<td>0%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>9%</td>
<td>19%</td>
<td>25%</td>
<td>27%</td>
</tr>
<tr>
<td>correct LDA\nJS</td>
<td>10%</td>
<td>8%</td>
<td>6%</td>
<td>7%</td>
<td>6%</td>
<td>6%</td>
<td>6%</td>
<td>8%</td>
</tr>
<tr>
<td>correct LDA\VSM</td>
<td>0%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>10%</td>
<td>17%</td>
<td>25%</td>
<td>26%</td>
</tr>
<tr>
<td>correct LDA\nVSM</td>
<td>11%</td>
<td>8%</td>
<td>6%</td>
<td>7%</td>
<td>6%</td>
<td>8%</td>
<td>7%</td>
<td>9%</td>
</tr>
<tr>
<td>correct LDA\LSI</td>
<td>13%</td>
<td>11%</td>
<td>9%</td>
<td>9%</td>
<td>15%</td>
<td>22%</td>
<td>28%</td>
<td>30%</td>
</tr>
<tr>
<td>correct LDA\nLSI</td>
<td>0%</td>
<td>7%</td>
<td>6%</td>
<td>7%</td>
<td>3%</td>
<td>5%</td>
<td>5%</td>
<td>7%</td>
</tr>
</tbody>
</table>
### Results for Overlap Metrics for eTour

<table>
<thead>
<tr>
<th></th>
<th>25</th>
<th>50</th>
<th>75</th>
<th>100</th>
<th>300</th>
<th>500</th>
<th>700</th>
<th>1K</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>correct LDA\JS</code></td>
<td>0%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>9%</td>
<td>19%</td>
<td>25%</td>
<td>27%</td>
</tr>
<tr>
<td><code>correct LDAnJS</code></td>
<td>10%</td>
<td>8%</td>
<td>6%</td>
<td>7%</td>
<td>6%</td>
<td>6%</td>
<td>6%</td>
<td>8%</td>
</tr>
<tr>
<td><code>correct LDA\VSM</code></td>
<td>0%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>10%</td>
<td>17%</td>
<td>25%</td>
<td>26%</td>
</tr>
<tr>
<td><code>correct LDAnVSM</code></td>
<td>11%</td>
<td>8%</td>
<td>6%</td>
<td>7%</td>
<td>6%</td>
<td>8%</td>
<td>7%</td>
<td>9%</td>
</tr>
<tr>
<td><code>correct LDA\LSI</code></td>
<td>13%</td>
<td>11%</td>
<td>9%</td>
<td>9%</td>
<td>15%</td>
<td>22%</td>
<td>28%</td>
<td>30%</td>
</tr>
<tr>
<td><code>correct LDAnLSI</code></td>
<td>0%</td>
<td>7%</td>
<td>6%</td>
<td>7%</td>
<td>3%</td>
<td>5%</td>
<td>5%</td>
<td>7%</td>
</tr>
</tbody>
</table>
Work in Progress

- More software systems (currently working with six datasets)
- Traceability links among different types of artifacts (use cases, design, source code and test cases)
- Impact of the number of dimensions (LSI) and the number of topics (LDA) on performance
- Impact of keyword filtering techniques (all terms vs. nouns)
- Combinations of different IR techniques
Conclusions

• JS, VSM, LSI are able to provide almost the same information when used for documentation-to-code traceability recovery.

• LDA is able to capture some information missed by VSM, LSI, and JS when used for recovering traceability links between code and documentation.

• LDA’s performance based on Hellinger Distance similarity measure is somewhat lower as compared to JS, VSM, and LSI
Thank you. Questions?

SEMERU @ William and Mary

http://www.cs.wm.edu/semeru/
denys@cs.wm.edu