Improving IR-based Traceability Recovery Using Smoothing Filters

Andrea De Lucia
Massimiliano Di Penta
Rocco Oliveto
Annibale Panichella
Sebastiano Panichella
Software traceability

“The degree to which a relationship can be established between two products of a software development process” [IEEE Glossary for Software Terminology]

- Important for:
  - program comprehension
  - requirement tracing
  - impact analysis
  - software reuse
  - ...

Up-to-date traceability links rarely exist → need to recover them
IR-based traceability recovery

Antoniol et al., 2002 (VSM+Probabilistic model)
Marcus and Maletic, 2003 (LSI)
Traditional IR vs. IR applied to Software Engineering

**Traditional IR**
- Deals with heterogeneous documents for what concerns:
  - Linguistic choices
  - Syntax
  - Semantics
- We just live with that differences

**IR applied to SE**
- We have sets of homogeneous documents for what concerns
  - Syntax, linguistic choices
- Examples:
  - Use cases, test documents, design documents follow a common template and contain recurrent words
Problem

Different kinds of software artifacts require specific preprocessing

Test case Change the date for a visit:
C51     Version: 0 02 000
Use case Satisfies the request to modify a visit for a patient
UcModVis
Priority High
....
Test description
Input Select a visit:
26/09/2003 11:00  First visit
Change: 03/10/2003 11:00
Oracle Invalid sequence: The system does not allow to change a booking
Coverage Valid classes: CE1  CE8  CE14  CE19  CE21
Invalid classes: None
**Problem**

- Different kinds of software artifacts require specific preprocessing

<table>
<thead>
<tr>
<th>Test case</th>
<th>Change the date for a visit:</th>
</tr>
</thead>
<tbody>
<tr>
<td>C51</td>
<td>Version: 0 02 000</td>
</tr>
</tbody>
</table>

**Use case**

- Satisfies the request to modify a visit for a patient

**UcModVis**

**Priority**

- High

**Test description**

**Input**

- Select a visit:
  - 26/09/2003 11:00 First visit
  - Change: 03/10/2003 11:00

**Oracle**

- Invalid sequence: The system does not allow to change a booking

**Coverage**

- Valid classes: CE1 CE8 CE14 CE19 CE21
- Invalid classes: None

Artifact-specific words do not bring useful information
A similar problem: image processing
Noisy images

Pixels with peaks of low color intensity

Noise

Pixels with peaks of high color intensity
Reducing noise using smoothing filters

Mean filter

\[ g(x, y) = \frac{1}{M} \sum_{f(n,m) \in S} f(n, m) \]
**Image vs. traceability noise**

**Image noise:**
- Pixels with high or low color intensity
- Pixels are position dependent

**Traceability noise:**
- Terms and linguistic patterns occurring in many artifacts of a given category
  - Use cases, test cases...
- Artifacts (columns) are position independent
### Representing the noise

#### Source Documents

<table>
<thead>
<tr>
<th>Word 1</th>
<th>V1,1</th>
<th>V1,2</th>
<th>V1,3</th>
<th>...</th>
<th>V1,k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word 2</td>
<td>V2,1</td>
<td>V2,2</td>
<td>V2,3</td>
<td>...</td>
<td>V2,k</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>Word n</td>
<td>Vn,1</td>
<td>Vn,2</td>
<td>Vn,3</td>
<td>...</td>
<td>Vn,k</td>
</tr>
</tbody>
</table>

Linguistic information strictly belonging to source documents

---

#### Target Documents

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>...</th>
<th>Tz</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1,1</td>
<td>V1,2</td>
<td>V1,3</td>
<td>...</td>
<td>V1,z</td>
</tr>
<tr>
<td>V2,1</td>
<td>V2,2</td>
<td>V2,3</td>
<td>...</td>
<td>V2,z</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vn,1</td>
<td>Vn,2</td>
<td>Vn,3</td>
<td>...</td>
<td>Vn,z</td>
</tr>
</tbody>
</table>

Linguistic information strictly belonging to target documents

---

**Common Information**

- **for source documents**
- **For target documents**
Representing the noise

### Source Documents

<table>
<thead>
<tr>
<th>word_1</th>
<th>v_{1,1}</th>
<th>v_{1,2}</th>
<th>v_{1,3}</th>
<th>...</th>
<th>v_{1,k}</th>
</tr>
</thead>
<tbody>
<tr>
<td>word_2</td>
<td>v_{2,1}</td>
<td>v_{2,2}</td>
<td>v_{2,3}</td>
<td>...</td>
<td>v_{2,k}</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>word_n</td>
<td>v_{n,1}</td>
<td>v_{n,2}</td>
<td>v_{n,3}</td>
<td>...</td>
<td>v_{n,k}</td>
</tr>
</tbody>
</table>

\[
S = \left[ \begin{array}{c}
\frac{1}{k} \sum_{j=1}^{k} v_{1,j} \\
\frac{1}{k} \sum_{j=1}^{k} v_{2,j} \\
\vdots \\
\frac{1}{k} \sum_{j=1}^{k} v_{n,j}
\end{array} \right]
\]

Mean source vector

Common Information for source documents

### Target Documents

<table>
<thead>
<tr>
<th>t_1</th>
<th>v_{1,1}</th>
<th>v_{1,2}</th>
<th>v_{1,3}</th>
<th>...</th>
<th>v_{1,z}</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_2</td>
<td>v_{2,1}</td>
<td>v_{2,2}</td>
<td>v_{2,3}</td>
<td>...</td>
<td>v_{2,z}</td>
</tr>
<tr>
<td>t_3</td>
<td>v_{3,1}</td>
<td>v_{3,2}</td>
<td>v_{3,3}</td>
<td>...</td>
<td>v_{3,z}</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>t_z</td>
<td>v_{z,1}</td>
<td>v_{z,2}</td>
<td>v_{z,3}</td>
<td>...</td>
<td>v_{z,z}</td>
</tr>
</tbody>
</table>

\[
T = \left[ \begin{array}{c}
\frac{1}{z} \sum_{j=k+1}^{m} v_{1,j} \\
\frac{1}{z} \sum_{j=k+1}^{m} v_{2,j} \\
\vdots \\
\frac{1}{z} \sum_{j=k+1}^{m} v_{n,j}
\end{array} \right]
\]

Mean target vector

Common Information for target documents

The Mean vectors are like the continuous component of a signal...
Representing the noise

Source Documents

\[
\begin{array}{cccccc}
S_1 & S_2 & S_3 & \ldots & S_k \\
\text{word}_1 & v_{1,1} & v_{1,2} & v_{1,3} & \ldots & v_{1,k} \\
\text{word}_2 & v_{2,1} & v_{2,2} & v_{2,3} & \ldots & v_{2,k} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\text{word}_n & v_{n,1} & v_{n,2} & v_{n,3} & \ldots & v_{n,k}
\end{array}
\]

Target Documents

\[
\begin{array}{cccccc}
t_1 & t_2 & t_3 & \ldots & t_z \\
v_{1,1} & v_{1,2} & v_{1,3} & \ldots & v_{1,z} \\
v_{2,1} & v_{2,2} & v_{2,3} & \ldots & v_{2,z} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
v_{n,1} & v_{n,2} & v_{n,3} & \ldots & v_{n,z}
\end{array}
\]

- \( S \)

(mean source vector)

\( S \)

Filtered Source Set

- \( T \)

(mean target vector)

\( T \)

Filtered Target Set
Empirical Study

- **Goal**: analyze the effect of smoothing filter
- **Purpose**: investigating how the filter affects traceability recovery
- **Quality focus**: traceability recovery performance
- **Perspective**:
  - **Researchers**: evaluating the novel technique
- **Context**: artifacts from two systems
  - EasyClinic and Pine
## Context

<table>
<thead>
<tr>
<th>EasyClinic</th>
<th>Pine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Medical doctor office management</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>Java</td>
</tr>
<tr>
<td><strong>Files/Classes</strong></td>
<td>37</td>
</tr>
<tr>
<td><strong>KLOC</strong></td>
<td>20</td>
</tr>
<tr>
<td><strong>Documents</strong></td>
<td>113</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>Italian</td>
</tr>
<tr>
<td><strong>Artifacts</strong></td>
<td>Use cases Interaction diagrams Source code Test cases</td>
</tr>
</tbody>
</table>
Research Questions and Factors

- **RQ1**: Does the smoothing filter improve the recovery performances of traceability recovery?
- **RQ2**: How effective is the smoothing filter in filtering out non-relevant words, as compared to stop word removal?

**Factors:**
- **Use of filter**: YES, NO
- **Technique**: VSM, LSI
- **Artifact**: Req., UC, Int. Diagrams, Code, TC
- **System**: Easyclinic, Pine
Analysis Method – RQ1

- Performances evaluated by precision and recall:

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

- We statistically compare the # of false positives of different methods for each correct link identified
  - Wilcoxon Rank Sum test
  - Cliff’ s delta effect size

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>
We replace stop word filtering by one of the following treatments:
1. Standard stop word removal
2. Manually customized stop word removal
3. Smoothing filter
4. Standard stop word removal + filter
5. Customized stop word removal + filter

...and compare the performances
Results
EasyClinic: Use cases into source (VSM)

Recall

Precision

Filtered

Not Filtered

[-60, -74]\% of false positives for recall<80%
EasyClinic: Use cases into source (LSI)

[-60, -77]% of false positives for recall < 80%
EasyClinic: Test cases into source (LSI)

Test cases are:
- Short documents
- Limited vocabulary
- Mostly consistent with source code
Pine: Use cases into requirements (LSI)

[-42, -62]% of false positives for recall < 80%
<table>
<thead>
<tr>
<th>Data set</th>
<th>Traced Artifacts</th>
<th>VSM</th>
<th>LSI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>p-value</td>
<td>Effect size</td>
</tr>
<tr>
<td>EasyClinic</td>
<td>UC→Code</td>
<td>&lt;0.01</td>
<td>0.50 (large)</td>
</tr>
<tr>
<td>Int. Diag. → Code</td>
<td>&lt;0.01</td>
<td>0.52 (large)</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td>TC → Code</td>
<td>1.00</td>
<td>- (negligible)</td>
<td>1.00</td>
</tr>
<tr>
<td>Pine</td>
<td>Req. → UC</td>
<td>&lt;0.01</td>
<td>0.58 (large)</td>
</tr>
</tbody>
</table>
## RQ2 – Summary of results

<table>
<thead>
<tr>
<th>Comparison</th>
<th>EasyClinic</th>
<th>Pine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UC→CC</td>
<td>ID→CC</td>
</tr>
<tr>
<td>Smoothing filter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard list</td>
<td>YES (small)</td>
<td>YES (small)</td>
</tr>
<tr>
<td>Cust. list</td>
<td>YES (small)</td>
<td>YES (small)</td>
</tr>
<tr>
<td>Standard list+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smoothing filter</td>
<td>YES (large)</td>
<td>YES (large)</td>
</tr>
<tr>
<td>Cust. list+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard list+</td>
<td>NO (small)</td>
<td>-</td>
</tr>
<tr>
<td>Smoothing filter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cist list</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Smoothing filter</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Link precision improvement

Login Patient vs. Person
Poor vocabulary overlap (10%)
Threats to validity

- **Construct validity**
  - Mainly related to our oracle
  - Provided by developers and for EasyClinic also peer-reviewed

- **Internal validity**
  - Improvements could be due to other reasons...
  - However, we compared different techniques (VSM, LSI)
  - The approach works well regardless of stop word removal, stemming and use of tf-idf

- **Conclusion validity**
  - Conclusions based on proper (non-parametric) statistics

- **External validity**
  - We considered systems with different characteristics and artifacts
  - ... but further studies are desirable
Conclusions

Representing the noise

Source Documents

Target Documents

EasyClinic: Use cases into source (LSI)

[-60, -77]% of false positives for recall<80%

EasyClinic: Test cases into source (LSI)

Test cases are:
- Short documents
- Limited vocabulary
- Mostly consistent with source code

RQ2 – Summary of results

<table>
<thead>
<tr>
<th>Comparison</th>
<th>EasyClinic</th>
<th>Pine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UC→CC</td>
<td>ID→CC</td>
</tr>
<tr>
<td>Smoothing filter</td>
<td>Standard list</td>
<td>YES (small)</td>
</tr>
<tr>
<td>Smoothing filter</td>
<td>Cust. list</td>
<td>YES (small)</td>
</tr>
<tr>
<td>Standard list+ Smoothing filter</td>
<td>Cust. list</td>
<td>YES (large)</td>
</tr>
<tr>
<td>Standard list+ Smoothing filter</td>
<td>Cust list + Smoothing filter</td>
<td>NO (small)</td>
</tr>
</tbody>
</table>
Work-in-progress

- Study replication
  - Different systems and artifacts
  - Use of relevance feedback
- More sophisticated smoothing technique
  - Non-linear filters
- Use in other applications of IR to software engineering
  - Impact analysis
  - Feature location