Using Semantics-Enabled Information Retrieval in Requirements Tracing

An Ongoing Experimental Investigation

Anas Mahmoud, Nan Niu*
Department of Computer Science and Engineering
Mississippi State University, USA
niu@cse.msstate.edu
http://www.cse.msstate.edu/~niu/
Outline

→ Introduction

→ Research Questions

→ Experimental Design

→ Preliminary Results

→ Concluding Remarks
What is “requirements tracing”?

“Software requirements traceability refers to the ability to describe and follow the life of a certain requirement, in both forward and backward directions.”

Gotel, O. and Finkelstein, A. ICRE’94

NASA-CM1-5.2.1.2: “The software module X shall schedule tasks based on priority, and shall time-slice tasks with the same priority.”
Why should we care about traceability?

→ Many standards consider it a quality indicator
  ± CMMI
  ± U.S. Federal Aviation Administration (FAA)
  ± ...

→ It is indispensable for carrying out many software engineering activities
  ± Verification
  ± Validation
  ± Change impact analysis
  ± System-level test coverage analysis
  ± Risk assessment
  ± ...

COMPSAC 2010
How to generate traceability links?

→ Manually: too tedious, expensive, and error-prone
  ­ NASA CM-1: 235 high-level requirements, 220 low-level (design) elements
  ­ RTM (requirements traceability matrix), in theory, contains 51,700 links

→ Automatic support: information retrieval (IR) techniques
  ­ Treat each high-level requirement as a query, then search the low-level elements and return those that are similar to the query
    ­ Much like Google returns results to a query
  ­ Many IR models have been employed in traceability links generation
    ­ Vector Space Model, Probabilistic Model, etc.
    ­ Treat software artifacts (requirements, code comments, etc.) as bags of weighted terms
  ­ Effectiveness
    ­ In order to recall at least 90% of the correct traceability links, precision levels of 10% to 50% can be achieved in most data sets
Room for improvement?

→ Semantic search
   ✏️ Seeks to improve search accuracy by understanding searcher intent and the contextual meaning of terms as they appear in the searchable data space to generate more relevant results (Wikipedia)
   ✏️ Check out this sequence of searches issued to Google
      ✏️ “computer science conference Korea”
      ✏️ “computer science conference Seoul Korea”
      ✏️ “software engineering conference Seoul Korea”
      ✏️ “software engineering conference Seoul Korea July 2010”
      The 4th hit is “COMPSAC 2010”

→ Motivation behind our investigation
   ✏️ Integrating semantics-enabled IR methods should generate more meaningful traceability links
Research Questions

→ Is it feasible to incorporate semantics in traceability links generation?

→ Is semantics-enabled IR more effective than syntactic IR in traceability links generation?
Outline

→ Introduction

→ Research Questions

→ Experimental Design

→ Preliminary Results

→ Concluding Remarks
Instrumentation

→ Data set: NASA CM-1 (scientific instrument)
  ➤ Industrial strength: RTM, in theory, contains $235 \times 220 = 51700$ links
  ➤ 361 true positives were validated (gold standard)

→ Tracing methods
  ➤ Given a high-level requirement $H$, rank every low-level element $L$
    according to its similarity to $H$

SYNTACTIC: Dice coefficient for two sets of words

$$S_1(H, L) = \frac{2 \cdot |H \cap L|}{(|H| + |L|)}$$

SEMANTIC: Cosine between two vectors (scaling each verb’s value by a ratio $\lambda = 2$)

$$S_2(H, L) = \frac{\sum_{i=1}^{N} h_i \cdot l_i}{\sqrt{\sum_{i=1}^{N} h_i^2 \sum_{i=1}^{N} l_i^2}}$$
Tool Support

<table>
<thead>
<tr>
<th>Id</th>
<th>Ref</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SRS5.1.1</td>
<td>The DPU-BOOT CSC shall execute when power is applied to the DPU.</td>
</tr>
<tr>
<td>1</td>
<td>SRS5.1.1</td>
<td>The DPU-BOOT CSC shall include a DRAM BIT consisting of the SYSTEM_BLOCK.</td>
</tr>
<tr>
<td>2</td>
<td>SRS5.1.1</td>
<td>The DPU-BOOT CSC shall create a record of failed DRAM blocks, BIT_DRAM.</td>
</tr>
<tr>
<td>3</td>
<td>SRS5.1.1</td>
<td>The DPU-BOOT CSC shall provide a monitor which accepts the BIT_DRAM.</td>
</tr>
<tr>
<td>4</td>
<td>SRS5.1.1</td>
<td>The Bootstrapping Monitor of DPU-BOOT CSC shall be initialized.</td>
</tr>
<tr>
<td>5</td>
<td>SRS5.1.1</td>
<td>The DPU-BOOT CSC shall increment the BIT_CNT parameter.</td>
</tr>
<tr>
<td>6</td>
<td>SRS5.1.1</td>
<td>The Bootstrapping Monitor of the DPU-BOOT CSC shall provide the BIT_DRAM.</td>
</tr>
</tbody>
</table>

The DPU-BOOT CSC shall create a record of failed DRAM blocks, BIT_DRAM, in the SYSTEM_BLOCK.

External Program Interfaces: Reads the BIT results from the location pointed to by pBitResults (which is assumed to be in EEPROM) into the BIT results data structure in DRAM.

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.075</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Count: 53, f: 0.138

Time: 00:00:00.125000
Hypotheses and Variables

→ Null Hypothesis \((H_0)\)
  - There is no significant effect of incorporating semantics in traceability links generation

→ Alternative Hypothesis \((H_1)\)
  - When generating traceability links, \textit{SEMANTIC} is significantly more effective than \textit{SYNTACTIC}

→ Independent Variable
  - The tracing method: \textit{SYNTACTIC} and \textit{SEMANTIC}

→ Controlled Variables
  - \textit{CM-1} data set; gold standard; tool support

→ Dependent Variables
  - Precision and recall
**Results**

Precision (accuracy) = \( \frac{C}{B} \)

Recall (coverage) = \( \frac{C}{A} \)

Differences are not statistically significant:
- Precision \( (F=4.213, p=0.063) \)
- Recall \( (F=0.880, p=0.367) \)
Outline

→ Introduction

→ Research Questions

→ Experimental Design

→ Preliminary Results

→ Concluding Remarks
Concluding Remarks

→ Research questions revisited
  ✗ Is it feasible to incorporate semantics in traceability links generation?
    ➢ Yes

  ✗ Is semantics-enabled IR more effective than syntactic IR in traceability links generation?
    ➢ No

→ Future work
  ✗ Investigate more and novel semantic IR techniques
  ✗ Perform sensitivity analysis of integrated tools
  ✗ Devise tracing-related metrics
  ✗ Consider human factors in the tracing process
Thank You!