

Traceability and Information Retrieval

Requirements Engineering II

Michał Kołtonik Serge Hänni

15.11.2010

Outline

- 1 Information Retrieval
 - Introduction
 - Algorithms
 - Quality Measurements
 - Information Retrieval in Requirements Engineering
 - Results

- 2 Human analysts in traceability

Information Retrieval

- given a document collection, find all documents matching a query
- documents and queries represented as vectors
- similarity of document and query: $d \cdot v = 3.4$

Algorithms

Vector Space Model

- based on angles between vectors

Latent Semantic Indexing

- connects terms that occur in similar context

Quality Measurements

recall

- no missing traces

precision

- no wrong traces

Information Retrieval in Requirements Engineering

- different levels
 - origin to requirement
 - requirement to code
- static
- traceability as a post-process

Results

High-level to low-level requirements (Hayes et al. 2006)

- recall 50%, precision 10%
- increased by human feedback up to recall 90%, precision 80%

Requirements to code (Antoniol et al. 2002)

- vector retrieval: recall 100%, precision 13.8%
- latent semantic indexing: recall 94%, precision 54%

Example

Requirements from high-level document being traced to elements in a lower level document

Step	Task
a	identify each requirement
b	assign a unique identifier to each requirement
c	for each high level requirement, locate all matching low level requirements
d	for each low level requirement, locate a parent requirement in the high level document
e	determine if each high level requirement has been completely satisfied
f	prepare a report that presents the traceability matrix
g	prepare a summary report that expresses the level of traceability of the document pair

Why an analyst needs IR tools

- all steps can be automated
- 20 high-level requirements $\times$ 20 low-level requirements = 400 candidate links to be considered (steps c-e)

Why traceability needs an analyst

- to provide the source data for automatic processes
- to provide feedback to IR engine (Rocchio algorithm)
- to evaluate candidate links (accept-reject)
- to decide if additional links should be sought
- **to give last say in all decisions**