SANAYOJAN: A Framework for Traceability Link Recovery between Use-Cases in Software Requirement Specification and Regulatory Documents

Ritika Jain, Ashish Sureka, Smita Gaisas
ritikajain7@gmail.com, ashish@iiitd.ac.in, smita.gaisas@tcs.com

Abstract

Our proposed solution approach consists of three phases:
1. Manual analysis and visual inspection of URS documents to automate the task of system use case extraction.
2. Implementation of patterns identified in previous step to automate traceability links recovery between system use cases and the regulations it needs to comply with.
3. Automated traceability links recovery between system use cases and the regulations it needs to comply with.

Solution Approach

Phase 1 – Features Identification

Spatial Features

Exploit physical representation of textual elements in URS documents. Identify Use cases written in tabular form.

Phase 2 - Implement Identified features

This phase consists of implementation of features identified in previous phase to automate the extraction of system use cases.

Phase 3 – Traceability Links Recovery

We define four categories of lexicon:
1. Use Case Specific Terms (UCST) – Process, User, Actor
2. Insurance-Domain Noun Terms (IDNT) – Agent, Claimant, Insured
3. Domain Agnostic Action Terms (DAAT) – Apply, Reject, Validate
4. User Interface Specific Terms (UIST) – Enter, Display, Button

Lexicon Features

We define two syntactic patterns:
Conditional use cases:
Example - If [IN] the policy is [VB] in trust then [RB] all the trustees must sign [VB] the death claim application.
Event – triggered use cases:
Example - Case must be referred at the early stage of the claim, after [MD] notification [VB] of death and on [IN] receipt of death certificate.

Synthetic Features

Miscellaneous Features – Spatial + Syntactic Features

We define two syntactic patterns:
Detecting use cases by exploiting different levels of text and syntactic patterns.

Detecting use cases written in parts

Experimental Dataset

69 real-projects URS Documents in Insurance domain from a large scale IT organization. Containing 1518 system use cases. 16 Regulatory Documents taken from IRDA with an average number of regulations to be 48 per document.

Performance Evaluation – System use case extraction

Applying the 10 identified features on the dataset, we get individual feature accuracy as shown in table below:

Performance Evaluation – Traceability Links Recovery

To measure the performance of the different comparers, we calculate Average Precision (AP) scores for these comparers which are as follows:

<table>
<thead>
<tr>
<th>Measure</th>
<th>AP Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>METEOR</td>
<td>0.47</td>
</tr>
<tr>
<td>Lexical</td>
<td>0.48</td>
</tr>
<tr>
<td>BLEU</td>
<td>0.24</td>
</tr>
<tr>
<td>Corley</td>
<td>0.45</td>
</tr>
<tr>
<td>LDA</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Conclusion

- An overall accuracy of 83.3% for the system use case extraction step.
- A combination of lexicon, syntactic and spatial feature can be used to identify system use cases.
- For traceability links recovery LDA proves to be the best formulating use case as query and regulations as documents.