Thresholding Strategy in Requirements Trace Retrieval

Xuchang Zou, Raffaella Settimi, Jane Cleland-Huang, Chuan Duan
Depaul University
School of Computer Science,
Telecommunication & Information System
Chicago, IL 60604
xzou@cti.depaul.edu

Abstract
This paper describes a categorized thresholding strategy to automatically discover the optimal thresholds during requirements tracing process. An objective function is proposed as a critical measurement in evaluating the performance of thresholding methods. Also, this paper reports early experimental results for this strategy and illustrates its satisfying performance over training dataset.

Keywords: Links Retrieval, Thresholding Strategy

1. Introduction

Information Retrieval techniques have been recently applied to the Requirements Engineering domain to model the requirements tracing process. Previous work in this area has demonstrated the use of information retrieval models, such as vector space, probabilistic inference network, etc, to retrieve traceability links either between higher and lower level requirements, or between requirements and other artifacts [1,2,3]. The retrieval algorithms compute similarity scores between pairs of artifacts. If the scores are above a certain threshold then a link is established, and will be returned to the user as a candidate link. Threshold values are typically set prior to the tracing process to determine candidate links. In this paper we discuss a categorized thresholding method to automate the discovery of the optimal threshold. The process is guided by an objective function that is defined as the critical measurement. Experiments show this strategy performs better than commonly used uncategorized thresholding method.

2. Guiding criteria for setting thresholds

Common thresholding strategies include [4,5]:
1) Ranking-based thresholding: Returns t top-ranking artifacts with respect to similarity scores (1 ≤ t ≤ n, where n is the number of potentially affected artifacts).
2) Proportion-based thresholding: Returns a percentage of artifacts in the list.
3) Constant thresholding: A fixed real value specified by the user.
Standard evaluation measurements for a thresholding strategy are recall and precision [6] defined as:

\[
\text{Recall} = \frac{\text{number of correctly retrieved links}}{\text{number of true links in the collection}}
\]

\[
\text{Precision} = \frac{\text{number of correctly retrieved links}}{\text{number of retrieved links}}
\]

The criterion for choosing an appropriate thresholding strategy depends on the trade-off between recall and precision. In contrast to a typical web-based search, the requirements engineer expects to retrieve as many true links as possible. A high-recall low-precision retrieval is preferable to one with low-recall and high-precision, as it is easier for the user to examine the subset of retrieved links, to filter out the unwanted ones, than to search through the entire document collection to find the true links.

In addition, we suggest an objective function “maximizing recall + precision, where recall > precision”. Thus, the optimal threshold will be selected as the value maximizing both recall and precision, for recall values larger than precision.

3. Categorized constant thresholds

A probabilistic network model is used to generate links between a query document and potentially affected artifacts. The probability of relevance of an artifact \(d_i\) with respect to a query \(q\) is defined as [2]:

\[
pr(d_j \mid q) = \left[ \sum_i pr(d_j \mid t_i) \cdot pr(q \mid t_i) \cdot pr(t_i) \right] / pr(q), \quad \text{where} \quad pr(t_i) = 1/n_i [\sum_i 1/n_i]^{-1} (n_i \text{ is the number of documents that contain the keyword } t_i), \quad pr(d_j \mid t_i) = \frac{\text{freq}(d_j, t_i)}{\sum_k \text{freq}(d_j, t_k)},
\]

\[
pr(q \mid t_i) = \frac{\text{freq}(q, t_i)}{\sum_k \text{freq}(q, t_k)} \quad \text{and} \quad pr(q) = \sum_i pr(q \mid t_i) \cdot p(t_i).
\]

Commonly used thresholding strategy assigns a global constant threshold \(T \in [0,1]\) to all potential links. Artifacts with \(pr(d_i \mid q) \geq T\) are considered as candidate links. Results from our experiments indicated large variance in the distributions of \(pr(d_i \mid q)\) for artifacts of different types. Given a specific query document, some categories have much higher probabilities than others. Therefore, it is intuitive to set up local optimal threshold for every type or category of artifact.

The algorithm used to discover approximate optimal thresholds on the training dataset is given as:

Categorized Optimal Thresholding(training dataset, Rt)

Input: Training dataset, which contains pairs of queries and potentially impacted artifacts with similarity scores.

Input: Rt, denotes target recall, which is established by the user.
For each category, do until termination condition is met:
1. Initialize T to a value T₀, where T₀∈[0,1]. We choose T₀=0.1 according to previous experiments.
2. Calculate the output recall R and precision P over all queries given threshold T.
   Store T if R and P satisfy the objective function (R+P is maximized compared to all previous output pairs of recall and precision, and R > P).
3. T←T+∆T, where ∆T=α*(R-Rt). α is a relatively small positive number to constraint the threshold adjustment. For the purposes of these experiment α=0.01.

Output: Optimal threshold T

Termination condition is:
1. Threshold adjustment ∆T is small enough (for instance, |∆T|≤0.0001, which occurs when output recall R is closely achieved towards target recall Rt ), or
2. T exceeds the bound (which occurs when target recall Rt was set too high to achieve).

An improvement is to give a possibly narrower boundary to T instead of 0 and 1. Given a specific query on a category we have a probability distribution scope [minimumProbability, maximumProbability]. When the values of minimum and maximum probability are close, this could reduce the effort of searching for optimal thresholds.

Initial experiments were conducted using the EBT (Event Based Traceability) dataset, which is a software tool for supporting evolutionary and speculative change [2]. Artifacts in the EBT dataset include 11 classes from one system level class diagram, 9 sequence diagram, 3 use case diagrams, 2 collaboration diagrams, 1 deployment diagram and others. Table 1 shows that categorized thresholds achieve satisfied performance both on recall and precision.

<table>
<thead>
<tr>
<th>Document type</th>
<th>Threshold</th>
<th>Recall</th>
<th>Precision</th>
<th>Recall + Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class diagram</td>
<td>0.0032</td>
<td>0.9545</td>
<td>0.1842</td>
<td>1.1388</td>
</tr>
<tr>
<td>Sequence diagram</td>
<td>0.0007</td>
<td>1.0000</td>
<td>0.1155</td>
<td>1.1155</td>
</tr>
<tr>
<td>Use case diagram</td>
<td>0.0013</td>
<td>1.0000</td>
<td>0.4167</td>
<td>1.4167</td>
</tr>
<tr>
<td>Collaboration diagram</td>
<td>0.0100</td>
<td>0.6000</td>
<td>0.6000</td>
<td>1.2000</td>
</tr>
<tr>
<td>Deployment diagram</td>
<td>0.0015</td>
<td>1.0000</td>
<td>0.5938</td>
<td>1.5938</td>
</tr>
<tr>
<td>Test case diagram</td>
<td>0.0159</td>
<td>0.8222</td>
<td>0.2587</td>
<td>1.2126</td>
</tr>
<tr>
<td>Java programming</td>
<td>0.0001</td>
<td>0.9582</td>
<td>0.0804</td>
<td>1.0396</td>
</tr>
</tbody>
</table>

4. Future work

The process of deriving optimal thresholds is performed on data from a training set, randomly chosen from the entire data set, and evaluated against a predefined traceability matrix. If training dataset or traceability matrix contains non-representative data, the derived threshold will be at risk of overfitting. If this occurs, the threshold might not perform as well on new data as it does on the training set.

To reduce this risk, a stratified strategy was applied to choose the training dataset, making it
most likely to represent the relevant and irrelevant distribution of the entire dataset. Another technique we are considering to avoid overfitting is to apply K-fold cross-validation. The entire dataset will be divided into K subsets of approximately equal size. One subset is left out of the training set and the remaining K-1 data is used as the testing set. After K times training and testing, the optimal threshold which yields the best average recall and precision will be derived.

In reality the training dataset in the requirements tracing problem is relatively small because manually building a large requirements traceability matrix for threshold evaluation is extremely time consuming. Training set selection strategies will therefore remain as future work in our research.

Reference