Abstract—Requirements tracing is a central activity for software systems quality management. However, in large-scale evolving systems, maintaining traceability information manually can become a tedious task. To address this problem, several dynamic techniques were introduced to provide automatic traceability links generation. These techniques are usually based on information retrieval (IR) methods which link different artifacts based on their syntactic information. This paper reports an ongoing experimental investigation of using semantics-enabled IR methods to generate traceability links. Our goal is to explore dynamic, accurate and conceptually rich ways to generate and maintain traceability information.

Keywords—Requirements tracing; information retrieval; dynamic link generation; semantic search

I. INTRODUCTION

Requirements traceability plays a vital role in ensuring that a software system meets its specifications. Traceability information provides the basis for many software engineering activities, including systems evolution, verification and validation, and change impact analysis [2].

Two ways of maintaining requirements traceability can be distinguished: static and dynamic. In the static approach, requirement traceability matrices (RTM) are constructed and maintained manually. However, in large-scale complex systems, where requirements evolve constantly over time, maintaining an up-to-date RM manually can become a tedious and expensive task [3]. To overcome this problem, several dynamic techniques were introduced to improve the process efficiency and minimize the manual effort. These techniques provide automatic support for generating traceability links at any point in the project life cycle.

Several information retrieval (IR) methods have been applied to facilitate dynamic link generation. Such methods try to recover traceability links by exploiting the software artifacts’ syntactic information. Previous work has shown that the effectiveness of IR methods in automating the tracing of textual requirements achieved high levels of recall (90%) at reasonable levels of precision (20%-40%) [2].

In their work, Antoniol et al. [4] have used two IR models: a probabilistic model and a vector space model, to recover traceability links between code and documentation. Their analysis showed that both models achieved 100% recall with almost the same number of documents retrieved and an average of 30% precision.

Lately, semantic search has become increasingly popular, especially in Web searching. Several studies (e.g., [5, 6]) have reported that integrating semantic knowledge into IR methods helps to improve the search process and generate more satisfying results. Semantics-enabled IR techniques try to understand the meanings in retrieved elements, rather than dealing with them as bags of weighted terms. The semantics-enabled approach is expected to generate more meaningful results by leveraging more advanced matching schemes.

In this paper, we describe an ongoing experimental investigation to assess the feasibility of integrating semantic knowledge into IR methods to generate traceability links. We aim to answer the following 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?

The results of our investigation are particularly valuable for practitioners who seek an accurate and conceptually rich way for obtaining and maintaining traceability information.

II. EXPERIMENTAL DESIGN

A. Instrumentation

We adopt NASA’s CM-1 data set [2] in our current experiment. The CM-1 data set consists of a complete requirements (high-level) document and a complete design (low-level) document for a NASA scientific instrument. The text of the documents has been altered by NASA prior to public release in order to hide the identity of the instrument. A typical requirement is one to two sentences in length. A typical design element is several paragraphs in length, with paragraphs averaging four to five sentences in length.

The CM-1 data set is appealing to our investigation for the following reasons. First, it is of industrial strength. The data set has 235 high-level requirements and 220 design elements. The RTM, in theory, contains 51,700 links. Second, the data set has been tested in previous work by mainly using syntactic IR methods [2]. As a result, the true positive traceability links have been identified and validated.
According to [2], there are 361 actual traces, which we use as the gold standard in our evaluation. Third, the data set is publicly accessible, making replication and cross-site validation possible.

The traceability links generation problem can be formulated as follows. Given a high-level requirement \( H \), a tracing method (an IR method in our context) ranks every low-level (design) element \( L \) according to its similarity to \( H \). We consider two IR methods in the current experiment.

- **Syntactic** treats \( H \) and \( L \) as two sets of words, and measures their similarity by Dice coefficient [7].

\[
S_2(H, L) = \frac{2 \cdot |H \cap L|}{(|H| + |L|)}
\]

- **Semantic** takes into account the meaning of \( H \) and \( L \), or part of \( H \) and \( L \). We extend the well-known vector space model, tf-idf (term frequency-inverse document frequency) [7], by performing part-of-speech tagging on \( H \) and \( L \). Previous work showed that verbs indicated the software’s functionalities [8]. For each verb in \( H \) and \( L \), we therefore scale its value in the vector space by a ratio \( \lambda = 2 \). This is equivalent to increase every verb’s weight by a scale \( \lambda \) in the tf-idf computation, which allows matches based on the verbs to obtain higher similarity ranks.

\[
S_1(H, L) = \frac{\sum_{i=1}^{N} h_i \cdot l_i}{\sqrt{\sum_{i=1}^{N} h_i^2 \sum_{i=1}^{N} l_i^2}}
\]

We fully implemented the above two IR methods. Our tool integrated OpenNLP (opennlp.sourceforge.net) for part-of-speech tagging. It is argued that state-of-the-art taggers like OpenNLP have the precision of about 97%, which makes them unlikely to become an extra error source [8]. However, sensitivity analysis on OpenNLP is beyond the scope of our current work.

### B. Hypothesis 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, **Semantic** is significantly more effective than **Syntactic**.

- **Independent Variable**: Our only independent variable is the IR method. It has two values: **Syntactic** and **Semantic**.

- **Controlled Variables**: We used the CM-1 data set and followed the gold standard in [2]. All traceability links were generated using our tool.

- **Dependent Variables**: Effectiveness was assessed by well-known IR metrics, **precision** and **recall** [7]. Precision measures accuracy and is defined as the proportion of generated traceability links which are correct. Recall measures coverage and is defined as the proportion of generated correct links to the total amount of all correct links.

![Figure 1. Traceability links generation effectiveness](image)

### III. Results

Figure 1 shows the precision and recall comparison between **Syntactic** and **Semantic**. We traced all 235 high-level requirements in CM-1. Threshold on the x-axis represents the cut-off value taken from the generated candidate links, e.g., 20% means using the top 20% of the candidate links for evaluation. We then plot the average precision and recall at each cut-off value.

**Semantic** generated more accurate traceability links as its precision outperformed that of **Syntactic**. The recall of **Semantic** to our surprise, was lower than that of **Syntactic**. Although differences existed in Figure 1, a repeated measure ANOVA (analysis of variance) at \( \alpha = 0.05 \) level revealed that the differences were not statistically significant: precision \( (F=4.213, p=0.063) \), recall \( (F=0.880, p=0.367) \). Thus, we failed to reject the null hypothesis.

### IV. Conclusions and Future Work

We have reported an ongoing experimental investigation of using semantics-enabled IR in dynamic traceability links generation. The **Semantic** model implemented in our tool demonstrates the feasibility of incorporating semantics in IR-based requirements tracing, though the preliminary results show the effect of incorporating semantics is not significant. Many new research avenues arise from our work reported here. It is important to explore more and novel semantic IR techniques and perform sensitivity analysis of integrated tools. It is also of great importance to scrutinize the effectiveness metrics (e.g., recall and F-measure) and devise tracing-related metrics (e.g., selectivity and lag [2]).

### References


