REquirements TRacing On target (RETRO) Enhanced with an Automated Thesaurus Builder:
An Empirical Study

Sandeep Pandanaboyana, Shreeram Sridharan, Jesse Yannelli, Jane Huffman Hayes
Computer Science
University of Kentucky
Lexington, Kentucky, USA
sandeeppandanaboyana@uky.edu, skera2@uky.edu, jesse.yannelli@uky.edu, hayes@cs.uky.edu

***Abstract*—Several techniques have been proposed to increase the performance of the tracing process, including use of a thesaurus. Some thesauri pre-exist and have been shown to improve the recall for some datasets. But the drawback is that they are manually generated by analysts based on study and analysis of the textual artifacts being traced. To alleviate that effort, we developed an application that accepts textual artifacts as input and generates a thesaurus dynamically, we call it Thesaurus Builder. We evaluated the performance of REquirements TRacing On target (RETRO) with a Thesaurus generated by Thesaurus Builder. We found that recall increased from 81.9% with no thesaurus to 87.18% when the dynamic thesaurus was used. We also found that Okapi weighting resulted in better recall and precision than TF-IDF weighting, but only precision was statistically significant.**

***Index Terms*—Traceability, RETRO, Automated Thesaurus, Ubiquitous Grand Challenge - Research Project 2**

I. INTRODUCTION

Tracing requirements has always been an important part of the software development life cycle. For example, it is essential to know that the product developed satisfies all its requirements. In addition, traceability information is required in order to perform impact analysis on any proposed changes. But requirements tracing is time consuming, prone to errors, and requires many mental comparisons, so we seek automation. RETRO [14] is one tool that automates tracing.

Requirements tracing consists of pairs of artifacts; we can think of them as high-level and low-level documents. For each element of a high-level document, a tracing tool searches the low-level document elements and generates a list of potential matches from the low-level documents called candidate links. A candidate link may either be correct, called a true link, or incorrect, called a false positive. Note that an analyst must vet the generated candidate links and may have to search for links not returned by the tool [3].

An automated tracing tool such as RETRO can be verified by using two common information retrieval (IR) measures: recall and precision. These are discussed in Section 2. In general, tracing tools are able to achieve high recall but retrieve many false positives also (low precision) [16, 17, 18].

In order to improve the performance (mainly precision) of tracing tools, researchers have undertaken a number of enhancements. For example, researchers have: used thesauri to detect synonyms and acronyms [6, 17], used phrasing [19], used neighborhoods of terms [16, 20], and used filtering based on similarity values [21, 3, 22, 17, 23, 24, 25, 19].

We focus on evaluating the performance of RETRO by taking into consideration two ways to improve performance: use of a thesaurus (our focus) and weighting scheme. We used Moderate Resolution Imaging Spectroradiometer (MODIS), a NASA open source dataset, in our study [9]. We applied the Vector Space Model (VSM) with two weighting options. We built a Thesaurus Builder tool and used it to generate a thesaurus for MODIS. We obtained recall and precision values for the different weighting options as well as for the different thesaurus options (none, standard thesaurus, dynamically generated thesaurus).

The paper is organized as follows. Section 2 presents background information. The evaluation study is discussed in Section 3. Results and analysis are presented in Section 4. Related work is discussed in Section 5. Section 6 provides conclusions and future work.

II. RESEARCH BACKGROUND

Information retrieval (IR) assists in finding information of interest from within a collection. In general, given a set of documents and a query, IR methods will help determine the set of documents in the collection that match the query [4]. We can frame requirements tracing as an IR problem as follows: we visualize the high level elements or requirements as queries and the low level elements as the document collection. There are several IR techniques in vogue, we considered the Vector Space Model. We applied two weighting options: Term Frequency-Inverse Document Frequency (TF-IDF) and Okapi.

The Vector Space Model represents each document and query as a vector of term weights. The TF-IDF weighing option looks at the term frequency (TF) and inverse document frequency (IDF) of a given term (for a given term or word: 1) how frequent is the term in the overall collection combined with 2) in how many documents does it occur). Okapi BM25 (simply called Okapi) is similar to TF-IDF but uses a term frequency dampening component [26]. Prior to building the corpus of terms and term weights, pre-processing may occur. For example, articles and conjunctions don't help in conveying meaning and are often discarded (called stop word removal). Also, terms may be stemmed to their grammatical root, this is called stemming.

Once the corpus and vectors have been built, the relevance of a given document to a query is expressed using a similarity measure or relevance weight; this is usually the cosine similarity of the angles of the two vectors.

The quality of IR methods [6] is determined based on the retrieved documents. Two metrics can be utilized for this purpose: precision and recall. Precision [6] is computed as the fraction of the relevant documents in the list of all documents returned by the IR method for a given query. Recall is the fraction of the retrieved relevant documents in the entire set of documents, retrieved and omitted, that are relevant to the query [6]. To measure the recall and precision of an IR method requires an answer set, or list of the true links that should be retrieved.

A high recall and low precision result from an IR method means that an analyst must perform the task of weeding out many false positives. On the other hand, a high precision and low recall result means that an analyst must search for and find potential links outside the list (and must realize that the links are missing). We would prefer to have high recall and high precision results, but this is rarely the case [15]. If given a choice of the aforementioned combinations, we therefore value high recall and low precision results over high precision and low recall results. This is because it has been shown to be easier to vet candidate links than to discover new links from scratch [4, 14].

Another option that we used to make the tracing process more effective is that of a thesaurus. We observed that there is great diversity in the documents to be traced since they were created by diverse organizations. For example, one organization may use the term "error" where another may choose to use the term "failure." An IR technique will not relate these two words of similar meaning. To overcome this issue, we used the Thesaurus option which is provided by RETRO. The thesaurus utilizes a set of triples (*v, a, w*), where *v* and *w* are the words that are relevant and *a* is a real number that shows how closely they are related [13].

A manually created thesaurus is available for MODIS, we refer to this as the Standard Thesaurus or ST. In addition, we developed a program to dynamically build a thesaurus (referred to as the Dynamic Thesaurus or DT) from the high and low level documents. We used the WordWeb thesaurus/dictionary SQL database for generating thesaurus entries [16].

Thesaurus Builder works as follows: initially, the application extracts all the words in a document; it first checks to see if a word is an acronym. We identify a word as an acronym if the word is represented as all capital letters. Then, we check to see if the word is an article or a conjunction; if it is, we add it to the Stopword file [11]. In addition, the analyst may add words to the Stopword file. Matching is then performed based on the following criteria: 1) the word is a synonym, or 2) the word is similar to a word in a low level document, or 3) the word is a type of another word (for example, printer is a type of hardware), or 4) the word belongs to the *types* set. To undertake these checks, Thesaurus Builder calls WWDevCOM3 and some of its methods such as LookupWord, WWDSynonyms, WWDSimilar, WWDTypes, WWDAllWordTypes, and WWDAllSenses. WWDevCOM3 determines if a word is a type of another word by using LookUp followed by GetRelated, this returns the text pairs as well as their relation type (such as "part of" or "type of"). Possible ambiguities (such as acknowledging that a printer is a type of hardware versus a type of worker) are prevented by looking at the definition sense. The pseudocode for Thesaurus Builder is shown in Figure 1. The startup screen is shown in Figure 2. The thesaurus tab is shown in Figure 3 and the unmatched word tab is shown in Figure 4.

```
Begin
    Get high and low document location
    Get location to save thesaurus
    If stopword file specified, get stopwords
    For all words in high and low documents that are not stopwords
    Use Wordweb to find synonyms
        If synonym found, add to thesaurus as similar
    Use Wordweb to get word type
        If conjunction or article, write to stopword list
    Use Wordweb to get type of
            If match found, add to thesaurus
    If acronym, add to acronym list
    If not matched in any of above checks, write to unmatched list
    Display all four lists
    Interact with user to add words to acronym or unmatched word list
     or to save thesaurus or stopwords
End
```

Fig. 1. Algorithm for Thesaurus Builder.

Note that at present we do not stem thesaurus entries. This could lead to false positives and/or missed matches if stemming is performed on the high and low level documents. The analyst can provide the thesaurus files (standard or application-generated) by enabling the Thesaurus option and providing the location for the file to be output. When RETRO is run with the Thesaurus option enabled, it performs one extra step: all the thesauri terms that are not in the corpus are made separate terms and are assigned weights.

III. EMPIRICAL ASSESSMENT

In this section, we present the design and hypotheses for our empirical study.

A. Study Design

We seek to determine whether the inclusion of a thesaurus improves the performance of RETRO. Further, we wonder if RETRO's performance improves when the Dynamic Thesaurus is used. We also examine the effect of weighting

option. In order to achieve our goals, we considered the following study design.

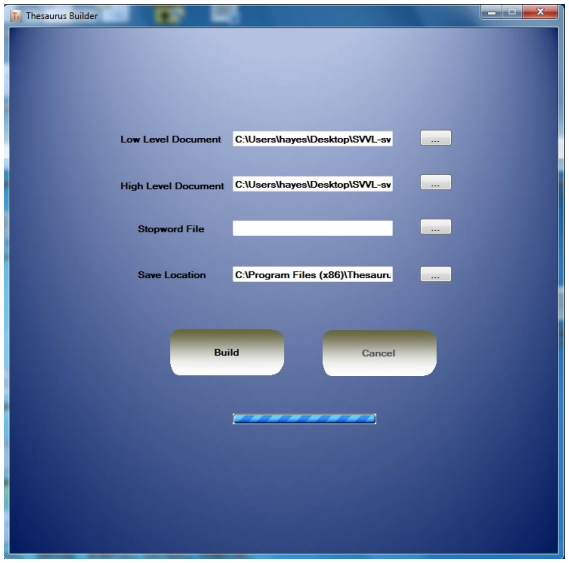

Fig. 2. Thesaurus Builder.

The dependent variables are recall and precision. The independent variables are weighting option (Okapi or TF-IDF) and thesaurus (no thesaurus (NT), standard thesaurus (ST), and dynamic thesaurus (DT)). In combining these, we therefore had TF-IDF with no thesaurus, TF-IDF with Standard Thesaurus, TF-IDF with Dynamic Thesaurus, Okapi with no thesaurus, Okapi with Standard Thesaurus, and Okapi with Dynamic Thesaurus. We ran the study on the MODIS dataset (available from coest.org). The dataset [9, 12] consists of 19 high level and 49 low-level requirements. The Vector Space Model (VSM) as described by Baeza-Yates [14] and implemented in RETRO [30] was applied.

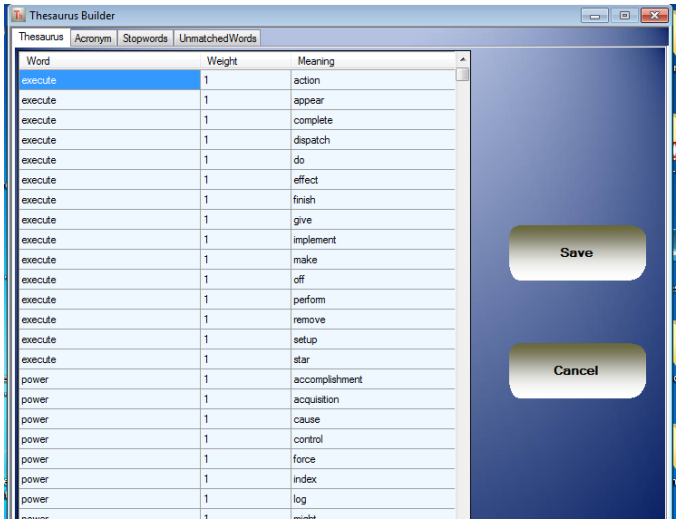

Fig. 3. The Thesaurus Tab of Thesaurus Builder.

B. Hypotheses

The hypotheses to be evaluated are as follows:

1) Null Hypotheses:

Modifying or including the thesaurus has no effect on the mean recall and mean precision, weighting option has no effect on mean recall and mean precision.

H_0-weight: $\mu R_{NTH\text{-}O} = \mu R_{STH\text{-}O}$, $\mu R_{NTH\text{-}O} = \mu R_{DTH\text{-}O}$
$\mu R_{NTH\text{-}TF} = \mu R_{STH\text{-}TF}$, $\mu R_{NTH\text{-}TF} = \mu R_{DTH\text{-}TF}$

H_0-thes: $\mu P_{NTH\text{-}O} = \mu P_{STH\text{-}O}$, $\mu P_{STH\text{-}O} = \mu P_{DTH\text{-}O}$
$\mu P_{NTH\text{-}TF} = \mu P_{STH\text{-}TF}$, $\mu P_{STH\text{-}TF} = \mu P_{DTH\text{-}TF}$

where
NTH – No Thesaurus option
STH – Standard Thesaurus option
DTH – Dynamic Thesaurus option
O – Okapi weighting option
TF- TF-IDF weighting option

2) Alternate Hypotheses:

Modifying or including the thesaurus has an effect on mean recall and mean precision, weighting option has an effect on mean recall and mean precision.

H_A-weight: $\mu R_{NTH\text{-}O} \neq \mu R_{STH\text{-}O}$, $\mu R_{NTH\text{-}O} \neq \mu R_{DTH\text{-}O}$
$\mu R_{NTH\text{-}TF} \neq \mu R_{STH\text{-}TF}$, $\mu R_{NTH\text{-}TF} \neq \mu R_{DTH\text{-}TF}$

H_A-thes: $\mu P_{NTH\text{-}O} \neq \mu P_{STH\text{-}O}$, $\mu P_{STH\text{-}O} \neq \mu P_{DTH\text{-}O}$
$\mu P_{NTH\text{-}TF} \neq \mu P_{STH\text{-}TF}$, $\mu P_{STH\text{-}TF} \neq \mu P_{DTH\text{-}TF}$

Though our alternative hypothesis is two sided, we expect to see better recall when the Dynamic Thesaurus is used, we have no pre-conceived notion on weighing option.

Note that we used the "two factor with no replication [2]" design, where modification of the thesaurus and weighting option are the factors and the treatments are the six methods we tested (TF-IDF+NTH, TFIDF+STH, TF-IDF+DTH, KE+NTH, KE+STH, KE+DTH). The data analysis tool in Excel for Analysis of Variance (ANOVA) for two factors with no replication was used to analyze the results [2].

IV. THREATS TO VALIDITY

To confirm the validity of our study, we consider four categories of threats to validity: internal validity, conclusion validity, construct validity, and external validity.

A. Threats to Internal Validity

"Threats to internal validity essentially constitutes of those factors that affect the value of the dependent variables in addition to the independent variables [2]." The primary threat to internal validity was that the generated thesaurus file might contain false matches which may lead to incorrect results – thus impacting recall and precision. To minimize this threat, we designed the Thesaurus Builder using the well vetted WordWeb dictionary/thesaurus.

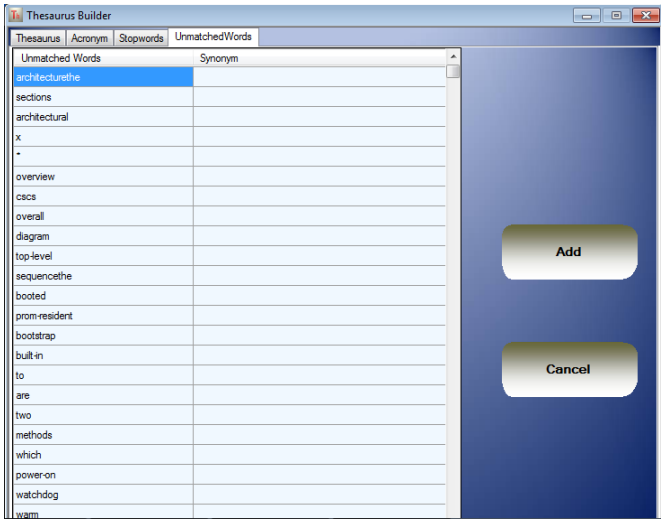

Fig. 4. The Unmatched Word Tab of Thesaurus Builder.

B. Threats to External Validity

"Threats to external validity are those that may limit the applicability of the experimental results to industry practice [2]." There were several threats to external validity. First, we used only one dataset. This dataset may not be representative of all datasets. Second, the dataset used is fairly small in size. However, the MODIS dataset is a real world dataset and is representative of the scientific instrument and space science domain.

Our study concentrates explicitly on the RETRO tracing tool. The study results may not be applicable to other tracing tools.

C. Threats to Conclusion Validity

"Threats to conclusion validity are concerned with issues that affect the ability to draw the correct conclusion about relations between treatment and the outcome of the experiment [2]." Our choice of statistical test is justified in the previous section and we also checked to ensure that the assumptions were met. Also, we repeated the study with the same setup and crosschecked the results.

D. Threats to Construct Validity

"Threats to construct validity refer to the extent to which the experiment setting actually reflects the construct under study [9]." Construct validity is concerned with generalizing the result of the experiment to the concept or theory behind the experiment.

In our study, the only threat to construct validity we came across was that of Mono-Operation bias. This arose due to the fact that we used a single data set (MODIS dataset). Since we used different methods to evaluate the tracing tool, though a single data set was used, we believe our study properly represented the construct.

V. STUDY OPERATION

The study proceeded as follows. Initially we ran RETRO on the dataset with no thesaurus file included. We selected the option of Vector Space Retrieval as the Information Retrieval technique and selected the weight option of TF-IDF. The tracing process was started by selecting the Trace All option in RETRO. The results of the tracing process were stored in the results.xml file. This file was provided as input to an application called Scantrace, which calculated the recall and precision. We then selected the weight option of Okapi with no thesaurus file included, performed the Trace All option, and calculated mean recall and precision.

In the second part of the study, we enabled the Thesaurus option and used the Standard Thesaurus for MODIS that had been manually created a number of years ago. We used Vector Space Retrieval and each of the weight options, and calculated recall and precision as described in the previous paragraph.

In the third part of the study, we used the Dynamic Thesaurus that was created by the Thesaurus Builder application. Again we used the Trace All option, used the two weight options (TF-IDF and Okapi), and calculated recall and precision.

VI. RESULTS AND ANALYSIS

The mean recall obtained for the different thesaurus and weighting options is shown in Table 1 and is depicted pictorially in Figure 5.

TABLE I. MEAN RECALL OBTAINED FOR THESAURUS AND WEIGHTING OPTIONS

	NTH	**STH**	**DTH**
VSR+TFIDF	75.6%	100%	80.48%
VSR+OKAPI	81.89%	100%	87.18%

As can be seen, the mean recall for STH for TF-IDF turned out to be 100%. For TF-IDF, the next highest mean recall was for DTH, at 80.48%. The mean recall for NTH with TF-IDF was 75.6%. The Okapi weighting option yielded higher recall, 100% for STH, 87.18% for DTH, and 81.89% for NTH.

We used Analysis of Variance with alpha of 0.05 to examine the recall values. Table 2 presents the descriptive data for this analysis while Table 3 presents the ANOVA results. In Table 2, the first column presents the item under study (weighting option or thesaurus option), the second column lists the element count, the third column presents the sum, the fourth column gives the average, and the final column provides the variance. In Table 3, the first column lists the source of variation, the second column lists the sums of the squares, the third column gives the degrees of freedom, the fourth column gives the mean squares, the F value is given in the fifth column, followed by p-value and F-critical in the final two columns.

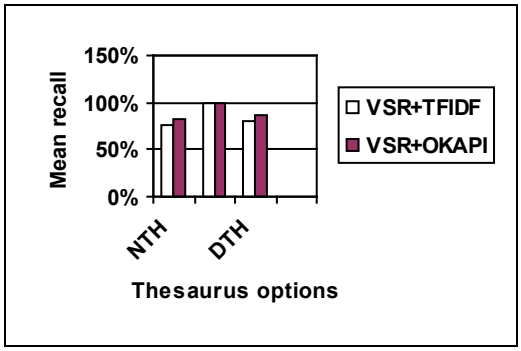

Fig. 5. Recall for thesaurus and weighting options.

As can be seen, the p-value for the weighting options was 0.1839. This is higher than 0.05, so there was no weight effect. The p-value for thesaurus factor, on the other hand, was 0.0278, showing an effect on recall. This confirms that the recall values obtained are different and there is an effect of thesaurus on the recall.

TABLE II. DESCRIPTIVE STATISTICS FOR MEAN RECALL FOR THESAURUS AND WEIGHTING OPTIONS

SUMMARY	*Count*	*Sum*	*Average*	*Variance*
TF-IDF	3	2.5608	0.8536	0.01667008
OKAPI	3	2.6907	0.8969	0.00867181
NTH	2	1.5749	0.78745	0.00197820
STH	2	2	1	0
DTH	2	1.6766	0.8383	0.0022445

TABLE III. P-VALUES FOR MEAN RECALL FOR THESAURUS AND WEIGHTING OPTIONS

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Weights	0.002	1	0.0028	3.98	0.1839	18.51
Thesauri	0.049	2	0.0246	34.94	0.0278	19
Error	0.001	2	0.0007			
Total	0.053	5				

The mean precision obtained for each of the methods when different thesaurus and weighting options were used is shown in Table 4 and is depicted pictorially in Figure 6.

It is clear that none of the precision values are "good" [17]. It should be noted that we used a filter of 10% (meaning we accepted links that had relevance weight of 0.10 or higher[1]). With TF-IDF, the mean precision for STH was 9.8%.

TABLE IV. PRECISION OBTAINED FOR THESAURUS AND WEIGHTING OPTIONS

	NTH	**STH**	**DTH**
VSR+TFIDF	7.63%	9.8%	5.73%
VSR+OKAPI	10.43%	13.39%	7.83%

The next highest mean precision was for NTH, at 7.63%. DTH was the worst of the three options with mean precision of 5.73%. OKAPI offered precision that was considerably higher: 13.39% for STH, 10.43% for NTH, and 7.83% for DTH.

TABLE V. DESCRIPTIVE STATISTICS FOR MEAN PRECISION FOR THESAURUS AND WEIGHTING OPTIONS

SUMMARY	*Count*	*Sum*	*Average*	*Variance*
TF-IDF	3	23.16	7.72	4.1473
OKAPI	3	31.6597	10.5532	7.7500138
NTH	2	18.0602	9.03010	3.92058802
STH	2	23.1966	11.5983	6.46776578
DTH	2	13.5629	6.78145	2.21111523

TABLE VI. P-VALUES FOR MEAN PRECISION FOR THESAURUS AND WEIGHTING OPTIONS

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Weights	12.040	1	12.04	43.111	0.0224	18.512
Thesauri	23.2360	2	11.61	41.597	0.0234	19
Error	0.55859	2	0.279			
Total	35.835	5				

Table 5 presents the descriptive data for this analysis. Table 6 presents the ANOVA results. The columns are the same as for Tables 2 and 3. It can be seen that the p-value for the weights was 0.0224, lower than the alpha of 0.05, showing that the weighting option did have an effect on the precision. Also, the p-value for thesaurus factor was 0.0234, showing an effect on precision. This confirms that the precision values obtained are different and there is an effect of thesaurus and weighting option on the precision.

[1] This value was selected based on visual examination of the matches.

VII. RELATED WORK

The use of a thesaurus for enhancing the precision and/or recall when undertaking requirements tracing is not uncommon. Hayes et al. found that using a thesaurus returned better results than Supertrace Plus (with a recall of 85.3% and precision of 40.6%) [6]. Sundaram et al. [13] used the vector space model (VSM) with term frequency-inverse document frequency (TF-IDF) weighting plus a thesaurus and achieved better precision than we achieved in the current study. Antoniol et al. [3] successfully used a thesaurus in tracing from code to documents. Specifically, they used the thesaurus to "help users to transform words into their roots. [3]"

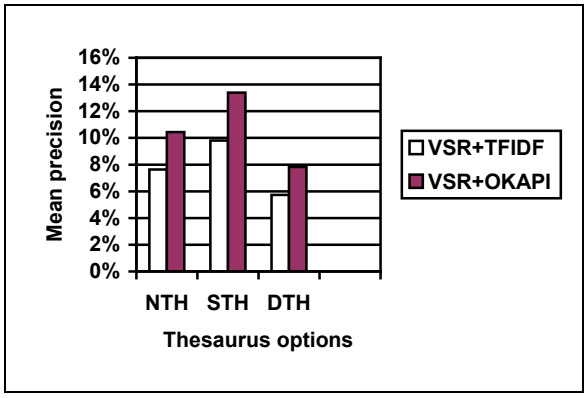

Fig. 6. Precision for thesaurus and weighting options.

Settimi et al. [25] used a general thesaurus in their study that focused on tracing from requirements to UML and from requirements to code. The thesaurus resulted in varying results, in some cases the thesaurus yielded slightly higher recall but lower precision, sometimes it yielded higher precision but slightly lower recall (this varied based on the document type, threshold, and weighting score scheme). In general, the thesaurus did not seem to impact results as much as did other factors [25].

In the above cases, a simple thesaurus was used. The users often had to manually create the thesaurus document, thus costing them much time and effort. One option for avoiding this is to generate a thesaurus from existing documentation. Hayes et al. [6] applied this idea and created the thesaurus by using an appendix document that was included with the requirements. This was faster than building a thesaurus from "scratch" but was still a manual process. Our approach differs in that we automatically generate the thesaurus with no user input and no required appendix or external document. Thesaurus Builder creates a dynamic thesaurus from the low level and high level documentation without needing input from the user. The WordNet lexical database [31] and its implementation (WordWeb) [32] is used to build the thesaurus. The use of a dynamic thesaurus decreased the time it took to use the tool.

VIII. CONCLUSIONS AND FUTURE WORK

From the above analysis, it is clear that the p-value obtained for the tests is less than the value of alpha with the exception of the weighting option for recall. This shows that there is an effect of thesaurus on both the recall and precision. Based on the results, we strongly reject the null hypothesis and accept the alternate hypothesis.

It can be noted that our Dynamic Thesaurus achieved greater recall compared to that obtained when no thesaurus was used. However, there was a decrease in precision when using the Dynamic Thesaurus. This can be explained by the fact that the Dynamic Thesaurus produced more false positive links than did the manually created thesaurus. But we can justify the use of the Dynamic Thesaurus over the Standard Thesaurus because it saves time.

Our future work can proceed in several directions. We would like to implement a dynamic adjustment feature in our application that allows the analyst to give us feedback on generated thesaurus entries. This would allow a more complete and effective thesaurus to be built. This feature could utilize standard Rochio feedback and allow analysts to accept or discard generated thesaurus entries. We would like to integrate Thesaurus Builder with RETRO and RETRO.NET. An additional future enhancement might be to use our Thesaurus Builder to build the initial thesaurus (DT) and then ask users to check the thesaurus and/or enhance it by spelling out acronyms, etc. (basically combining DT with ST). This would allow us to apply knowledge we have gained in interacting with analysts [27, 28, 29] vetting candidate links to the vetting of a thesaurus list. A minor future improvement will be to stem the thesaurus entry words before matching high level and low level documents.

ACKNOWLEDGMENTS

We thank Jody Larsen for his assistance with RETRO. We thank Wenbin Li for his assistance. We thank Wei-Keat Kong for CANDLE. We thank NASA for the MODIS dataset.

REFERENCES

[1] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "Introduction to Information Retrieval," Cambridge University Press, 2007.

[2] Clases Wohlin, Per Runeson, Martin Host, Magnus C. Ohlsson, Bjorn Regnell, Anders Wesslen, "Experimentation in Software Engineering: An Introduction," November 1999.

[3] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E.Merlo, "Recovering Traceability Links between Code and Documentation". IEEE Transactions on Software Engineering, Volume 28, No. 10, October 2002, 970-983.

[4] Jane Huffman Hayes, Dekhtyar, A., Sundaram, S.K., and Sarah Howard, "Helping Analysts Trace Requirements: An Objective Look," in Proceedings, 12th International Requirements Engineering Conference (RE 2004), pp. 249-261, September 2004, Kyoto, Japan.

[5] Jane Huffman Hayes, Alex Dekhtyar, "A Framework for Comparing Requirements Tracing Experiments". International Journal of Software Engineering and Knowledge Engineering 15(5): 751-782 (2005).

[6] Jane Huffman Hayes, Alexander Dekhtyar, James Osbourne, "Improving Requirements Tracing via Information Retrieval," in Proceedings of the International Conference on Requirements Engineering, Monterey, California, September 2003.

[7] J.Matthias, Requirements tracing communications of the ACM, 41 (12), 1998.

[8] Kitchenham, B.A., et al, "Preliminary guidelines for empirical research in software engineering". IEEE Transactions on software Engineering, 28(8), 2002.

[9] MODIS Science Data Processing Software Requirements Specification Version 2, SDST089, GSFC SBRS, November 10, 1997.

[10] Muhammad Ali Babar, Barbara Kitchen ham, Ross Jeffery, "Distributed Versus Face-to-Face Meetings for Architecture Evaluation: A Controlled Experiment". Proceedings of the 2006, ACM/IEEE international symposium on International symposium on empirical software engineering.

[11]Reference for stop words: http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words.

[12] Requirements Specification, SDST-0591, GSFC SBRS, September 11, 1997.

[13] Senthil Sundaram, Jane Huffman Hayes, Alex Dekhtyar, "Baselines in Requirements Tracing," Proceedings of Workshop on Predictive Models of Software Engineering (PROMISE), associated with ICSE 2005, St. Louis, MO, May 2005.

[14] Jane Huffman Hayes, Alex Dekhtyar, Senthil Sundaram, Ashlee Holbrook, Sravanthi Vadlamudi, Alain April, "Requirements Tracing On target (RETRO): Improving Software Maintenance through Traceability Recovery," Innovations in Systems and Software Engineering: A NASA Journal (ISSE) 3(3): 193-202 (2007).

[15] Jane Cleland-Huang (Editor), Orlena Gotel (Editor), Andrea Zisman (Editor), Software and Systems Traceability, Springer, 2012.

[16] Hakim Sultanov, Jane Huffman Hayes, and Wei-Keat Kong, "Application of Swarm Techniques to Requirement Tracing," Special Issue of Requirements Engineering Journal, Best 5 Papers of Requirements Engineering Conference 2010, Volume 16, Issue 3 (2011), Page 209-226.

[17] Jane Huffman Hayes, Alex Dekhtyar, Senthil Karthikeyan Sundaram, "Advancing Candidate Link Generation for Requirements Tracing: The Study of Methods," IEEE Transactions on Software Engineering, Vol. 32, No. 1, pp. 4-19, January 2006.

[18] Jane Cleland-Huang, Brian Berenbach, Stephen Clark, Raffaella Settimi, Eli Romanova: Best Practices for Automated Traceability. IEEE Computer 40(6): 27-35 (2007).

[19] Xuchang Zou, Raffaella Settimi, Jane Cleland-Huang: Improving automated requirements trace retrieval: a study of term-based enhancement methods. Empirical Software Engineering (ESE) 15(2):119-146 (2010).

[20] Hakim Sultanov, Jane Huffman Hayes, "Application of Swarm Techniques to Requirements Engineering: Requirements Tracing," Proceedings of IEEE International Conference on Requirements Engineering (RE), September 2010, Sydney, Australia, RE 2010: 211-220.

[21] A Abadi, M Nisenson, Y Simionovici, A Traceability Technique for Specifications, in Proceedings of the 16th IEEE International Conference on Program Comprehension, ICPC 2008, pp. 103—112.

[22] Andrea De Lucia, Fausto Fasano, Rocco Oliveto, and Genoveffa Tortora. 2007. Recovering traceability links in software artifact management systems using information retrieval methods. ACM Trans. Softw. Eng. Methodol. 16, 4, Article 13 (September 2007).

[23] Marco Lormans, Arie van Deursen, Hans-Gerhard Groß: An industrial case study in reconstructing requirements views. Empirical Software Engineering 13(6): 727-760 (2008).

[24] Andrian Marcus and Jonathan I. Maletic. 2003. Recovering documentation-to-source-code traceability links using latent semantic indexing. In Proceedings of the 25th International Conference on Software Engineering (ICSE '03). IEEE Computer Society, Washington, DC, USA, 125-135.

[25] Raffaella Settimi, Jane Cleland-Huang, Oussama Ben Khadra, Jigar Mody, Wiktor Lukasik, and Chris DePalma. 2004. Supporting Software Evolution through Dynamically Retrieving Traces to UML Artifacts. In Proceedings of the Principles of Software Evolution, 7th International Workshop (IWPSE '04). IEEE Computer Society, Washington, DC, USA, 49-54.

[26] John S. Whissell and Charles L. Clarke. 2011. Improving document clustering using Okapi BM25 feature weighting. Inf. Retr. 14, 5 (October 2011), 466-487.

[27] Kong, WK, Hayes, JH, Dekhtyar, A, Dekhtyar, O, "Process Improvement for Traceability: A Study of Human Fallibility," in Proceedings of the IEEE International Conference on Requirements Engineering (RE) 2012.

[28] Alex Dekhtyar, Olga Dekhtyar, Jeff Holden, Jane Huffman Hayes, David Cuddeback and Wei-Keat Kong, "On Human Analyst Performance in Assisted Requirements Tracing: Statistical Analysis," to appear in Proceedings of IEEE International Conference on Requirements Engineering (RE) 2011, Trento, Italy.

[29] David Cuddeback, Alex Dekhtyar, Jane Huffman Hayes, "Automated Requirements Traceability: the Study of Human Analysts," Proceedings of IEEE International Conference on Requirements Engineering (RE), September 2010, Sydney, Australia, RE 2010: 231-240.

[30] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. 1999. Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[31] George A. Miller (1995). WordNet: A Lexical Database for English, Communications of the ACM, Vol. 38, No. 11: 39-41.

[32] Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.