IR based Traceability Link Recovery Method Mining

Takeyuki Ueda, Shinpei Ogata, Haruhiko Kaiya, Kenji Kaijiri
Shinshu University
Nagano, Japan
Email: ueda, ogata, kaiya, kaijiri@cs.shinshu-u.ac.jp

***Abstract*—Traceability link recovery is an important process in software development, and several researches are done, but the generality of adequate methods is not considered. The target of traceability link recovery includes several kinds of documents, so the adequacy of recovery methods depends on the characteristics of these documents, for example, an average similarity, a kind of document pair, document size, and so on. We propose the traceability link recovery method mining, which identifies a kind of adequate recovery method based on the characteristics of target documents by using knowledge base consisting of (a method, characteristics, and performance). This knowledge base shows which pair (a method, characteristics) is good at performance. Our target traceability link recovery method is IR based method, which is major method of automated traceability recovery. Some experiments based on the traceability reference data sets are done and the potential of our method is shown.**

Keywords-traceability; mining; information retrieval

I. Introduction

In experimental software engineering, especially in estimating software quality factors, several methods are proposed and the effectiveness of these methods is evaluated. However external validity of these evaluations is not validated, because there are a variety of target domains and these objective artifacts have a variety of characteristics. Therefore an experimental result for some artifacts is not applied to another artifacts, but validation of external validity is indispensable for application to real software artifacts. Wohlin [1] said that threats to external validity are conditions that limit our ability to generalize the results of our experiment to industrial practice.

Zhimin [2] proposed a method mining technique for error prone module prediction. In error prone module prediction, predictors are constructed based on training data by using some mining algorithm and software metrics, so these factors must be determined before applying prediction. There are many researches about this domain and several proposals for adequate algorithms, metrics, and training data have been done, but these results are not generally validated, that is, threats to validity about external validity is not solved. Zhimin [2] constructed a knowledge base about the adequate set for prediction, and by using this knowledge base, the adequate algorithm and the training data are estimated. Zhimin's main idea is to reuse the performance data based on the similarity of characteristics.

On the other hand, traceability link recovery is the important research topic in software maintenance. Traceability link is the relation between software artifacts, for example, requirement statements and design statements, design statements and source code, functional requirements and nonfunctional requirement. These links may be missed during development, so their reestablishment is needed. This reestablishment is the objective of traceability link recovery. Asuncion [3] categorizes traceability link recovery into two categories: retrospective traceability and prospective traceability. The former is automated approach and Information Retrieval(IR) based method is the representative and it recovers the traceability link based on the document automatically. The latter is semi automated or manual approach. The retrospective traceability is more available than the prospective traceability, but it is not so precise. So improvement and guarantee of preciseness are the main research topic in IR based traceability link recovery. There are many researches [4]–[12], and several methods are proposed, but the best method is not identified. The adequacy of method is considered to be dependent on the characteristics of target document pair.

In this paper, we propose the application of Zhimin's method to IR based traceability link recovery. In this case, the triple (a method instance, characteristics, performance) is training data, the tuple (a method instance, characteristics) is test data. A method instance is a certain combination of a variety of traceability link recovery techniques. Performance is the measure of how good this method instance is. Characteristics are factors which may affect the performance. We suppose that an average similarity, a kind of document pair, document size, and so on are candidates for the characteristics.

We also propose new cosine similarity, which reflects link semantics. Several experiments using reference data set provided in CoEST [13] are done and the effectiveness of new cosine similarity and our mining method is shown. Our main contributions are as follows:

- A traceability recovery method mining method is proposed.
- The accuracy of the selected traceability link recovery method is assured.
- Asymmetrical cosine similarity is proposed.

In Section 2, we describe the traceability link recovery problem and in Section 3, we discuss the application of the method mining model to traceability domain. In Section 4, we describe the experiment and the result, and show the effectiveness of this method. In Section 5, we consider threats to validity, and in Section 6, we compare with related researches. We conclude our paper and consider future work in Section 7.

II. IR BASED TRACEABILITY LINK RECOVERY

Traceability link means the relation between some software components within several software documents. A software project has several kinds of documents and these documents consist of several components; for example, a software project has requirement document, design document, source code and test document, and each of these documents have their own components (the requirement document consists of many requirement statements and the source code consists of many class files).

There are several approaches about traceability link recovery [4] and the most possible method is IR based method. In IR based method, each component is modeled as a term vector, and the similarity between components is measured by using the cosine similarity of these term vectors. Traceability link will be identified by using these similarity values. These term vectors are aggregated into a term document matrix. There are several variations of construction methods of a term document matrix:

1) Term extraction and preprocessing: Stemming, stop word, Camel case
2) Kind of value for term vector: True/False, frequency, term frequency-inverse document frequency(TF-IDF)
3) Link candidate judgment method by using similarity value: threshold value (top n%) or rank (top n pairs)
4) Modification of term document matrix: Latent Semantic Indexing(LSI)

These are traditional variations for IR based method. We consider a further variation, which is specific for software link recovery, asymmetric cosine similarity. The cosine similarity treats each component symmetrically, but several kinds of relations are proposed in software traceability [14], [15], for example, Ramesh [14] proposed the following four kinds of link:

- Satisfaction link
- Evolution link
- Rationale link
- Dependency link

These relations are not necessarily symmetric, so we define asymmetrical cosine similarity as follows:

$$X \times Y/(|X| * |Y|) \qquad (1)$$

if $X_i == 0$ then the corresponding Y_i is not considered.
where X and Y are term vectors and X=(X_1,,,X_n) , Y=(Y_1,,,Y_n)

Asymmetrical similarity considers only how much X is covered by Y, for example, X=(0,0,1,1,0) and Y=(1,0,1,1,0), then symmetrical similarity(X,Y) = 2/(sqrt(2)*sqrt(3))=0.816, and asymmetrical similarity(X,Y)=2/(sqrt(2)*sqrt(2))=1.0.

We apply this variation as the 5th variation.

There are several other variations, for example, the granularity of components, ontology, etc. The treatment of these alternatives is the future research theme.

Each traceability recovery method selects one alternative from each variation. The following is an example:

- stemming is used
- stop word is eliminated
- camel case word is decoupled
- value of term document matrix is TF-IDF
- link candidate is judged with threshold value (0.3)
- LSI is applied
- symmetrical cosine similarity is applied

We call these alternatives as method instances. Selection of adequate method instances for each data is the main target of our research.

We afford the following research questions:

- RQ1: Is it possible to identify the adequate method instance for each project data?
- RQ2: Is it possible to assure the accuracy of the selected method instance?
- RQ3: Is the asymmetric similarity is effective for traceability link recovery?

III. TRACEABILITY LINK RECOVERY METHOD MINING

We show the traceability link recovery method mining in Figure 1. In order to do traceability link recovery, an adequate method instance has to be identified, and we supposed that the adequacy is dependent on the characteristics of target documents, so it must be possible to identify the adequate method instance candidate by using these characteristics. We use data mining approach proposed by Zhimin [2] for this identification. For this purpose we need to select the adequate characteristics. In this paper, we use CoEST [13] data set as a reference data set. Each document consists of two component sets and link between these component. We select the following characteristics which can be extracted from these documents:

- Average similarity
- Number of components
- Total term count
- Used language
- Type of document relation

These characteristics may be insufficient and the adequacy needs to be further considered.

As shown in Figure 1, the following training data and test data are needed:

- Training data: (a method instance, characteristics, performance)
- Test data : (a method instance, characteristics)

Performance is transformed into true/false value based on the traceability link criterion which is defined by using precision and recall. There are a few reports about the traceability link criterion. Hayes [16] described that adequate recall value is from 60 to 69% and adequate precision value is from 20 to

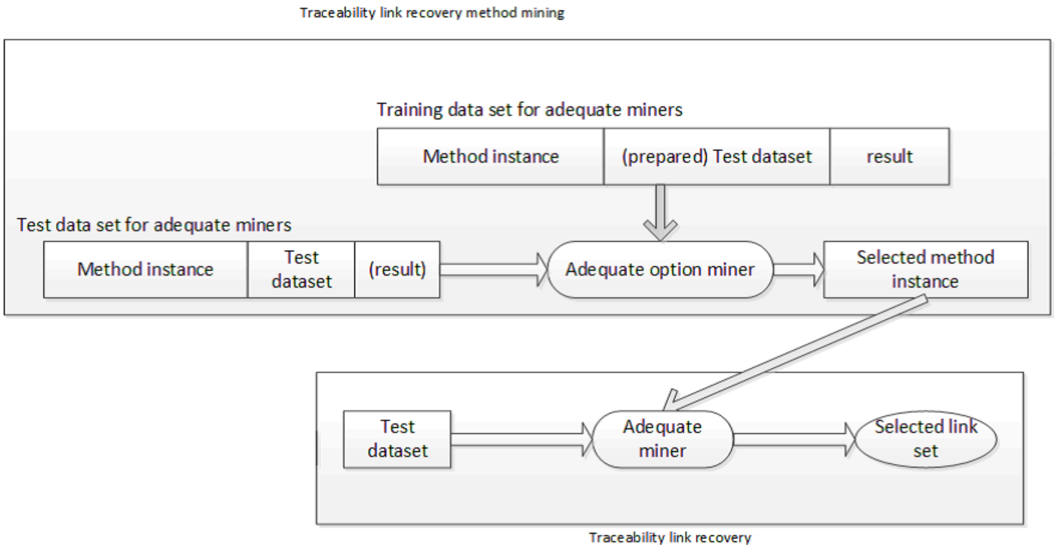

Fig. 1. Outline of traceability link miner selection

29% based on enterprise experiences. We use these values as objective values, and adjust these values based on conditions.

We show the procedure for traceability link recovery method mining.

1) For each method instance and each data in reference data set
   a) Do traceability link recovery
   b) Based on the traceability link criterion, judge the adequacy of recovery result
2) Training data is generated
3) Do method mining by using training data
4) If the identification performance is below the identification criterion, lower the traceability link criterion and repeat from step 2
5) By using this data, we construct a method miner
6) Construct test data
7) Do method mining by using (method minor, test data) pair
8) If no method is mined, lower the traceability link criterion and repeat from step 2
9) Do traceability link recovery by using the selected method instance

Step 3 and 4 are for the examination of adequacy of the selected training data. If the traceability link criterion becomes too small in step 8, it means that reasonable traceability link recovery is impossible for this test data. There are the following reasons:

- Training data is inadequate for test data.
- A variety of method instances are insufficient.
- Method mining method is insufficient.
- The quality of the document is too low.

We define two criteria:

- Traceability link criterion
  There must be adequate accuracy value in traceability link recovery, and it is the traceability link criterion. We use Hayes proposed value for this criterion, that is, $precision > 0.3$ and $recall > 0.7$, but there are several cases for this adequacy, so we may adjust this criterion. We further call the pair (precision, recall), which is the result of traceability link recovery, as traceability link recovery performance. We call a method instance, which satisfies this criterion, as a candidate method instance.
- Identification criterion
  The selected method instance needs to assure the satisfaction of the given traceability link criterion. We use the following precision for this purpose

$$\frac{(\mid CMIS \mid) \cap (\mid SMIS \mid)}{\mid SMIS \mid} \tag{2}$$

where CMIS is a set of candidate method instances, and SMIS is a set of selected method instances. This criterion is computed for each project, that is, a pair of documents. For example, if the identification criterion is 0.8, then 80% of the selected method instances are supposed to satisfy the traceability link criterion. We call the result of traceability link recovery method mining as identification performance. It only shows possibility, that is, the satisfaction of the traceability link criterion is only exemplified using training data, so if the test data has similarity with the training data, this possibility is high, but if the test data has no similarity, then this possibility is low.

IV. Experiments

We used the subset of the reference data set provided by CoEST [13]. The details of the used data sets are shown in Table I. The third column presents the numbers of each component (source component and destination component), the fourth column contains the average number of correct

TABLE I
REFERENCE DATA SET

project	document pair	# of components	average # of links	# of candidate method instances
eAnci	UC_CC	140,55	3.10	0
Gantt	high_low	17,69	4.00	0
SMOS	UC_CC	67, 100	15.58	0
WV_CCHIT	Requirements-Regulatory code	116, 1064	5.06	0
EasyClinic	CC_TC	47,63	4.34	0
EasyClinic	ID_CC	20, 47	3.45	89
EasyClinic	ID_TC	20, 63	4.15	37
EasyClinic	ID_UC	20, 30	1.30	13
EasyClinic	TC_CC	63, 47	3.24	0
EasyClinic	UC_CC	30, 47	3.10	47
EasyClinic	UC_ID	30, 20	0.87	15
EasyClinic	UC_TC	30, 63	2.10	16
Waterloo grp01	high,low	58, 26	0.52	0
Waterloo grp02	high,low	42, 13	1.24	0
Waterloo grp03	high,low	70, 28	1.34	0
Waterloo grp05	high,low	54, 30	0.87	2
Waterloo grp06	high,low	39, 21	1.41	0
Waterloo grp08	high,low	85, 22	1.08	0
Waterloo grp09	high,low	30, 19	1.77	0
Waterloo grp10	high,low	76, 8	0.91	0
Waterloo grp11	high,low	79, 9	0.89	0
Waterloo grp13	high,low	43, 8	0.72	0
Waterloo grp14	high,low	46, 5	0.72	24
Waterloo grp15	high,low	69, 27	1.35	0
Waterloo grp17	high,low	57, 7	0.89	0
Waterloo grp18	high,low	53, 8	0.66	78
Waterloo grp19	high,low	61, 15	2.03	0
Waterloo grp20	high,low	93, 14	1.49	0
Waterloo grp21	high,low	36, 26	1.14	25
Waterloo grp23	high,low	32, 20	1.06	12
Waterloo grp24	high,low	51, 29	1.10	0
Waterloo grp30	high,low	48, 20	0.73	0
Waterloo grp32	high,low	86, 21	1.57	0
Waterloo grp33	high,low	65, 11	0.94	0
Waterloo grp34	high,low	28, 16	0.64	0

links, and the fifth column contains the number of method instances, which identify link candidates with $precision > 0.3$ and $recall > 0.7$.

The detailed experimental results are too large, so we store the results in http://cwww.cs.shinshu-u.ac.jp/ICSEA/ and show only the summarized results.

We used seven threshold values (10%, 20%, 30%, 40%, 50%, 60%, 70%) and four rank values (5, 10, 15, 20) with the five kinds of variation described in Section II, so the total number of method instances is 1056.

We did three experiments by using the data mining tool Weka [17] in order to evaluate the effectiveness of our method.

A. Experiment 1: Traceability link recovery evaluation for the 35 data

We did 1056 runs (method instances) for each data: total 36960 (1056×35) runs. Each run calculates candidate link set and the accuracy is evaluated by using the given answer link set.

TABLE II
THE RESULT OF TRACEABILITY LINK RECOVERY

	precision	recall	f-measure
average	0.196	0.520	0.171
standard deviation	0.228	0.376	0.128

We evaluated the effectiveness of each method instance for each data. We show the number of candidate method instances, whose traceability link performance satisfies the traceability link criterion, in Table I and the statistic values in Table II. In this experiment, the traceability link criterion is $precision > 0.3$ and $recall > 0.7$. The deviation of the number of candidate method instances are large, that is, in 24 out of 35 data, the number of candidate method instances is zero, but EasyClinic ID_CC has 89 candidate method instances and Waterloo grp18 has 78 candidate method instances. The standard deviations of performance values (precision, recall, f-measure) are also large, so there must be adequacy of method instances for each data set. We show the scatter plot diagram in Figure 2. The horizontal axis is the index of each method instance and the vertical axis is the number of occurrences in the top 3 method instances with f-measure.

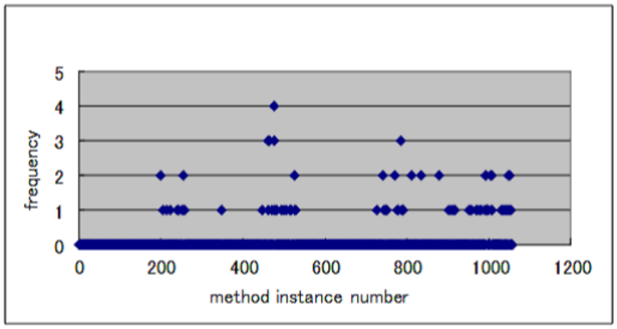

Fig. 2. Scatter plot diagram top 3 method instances

This figure also shows that there is no unique method instance, which has the best performance, that is, plots are dispersed. The O and P column of exp1.xlsx in http://cwww.cs.shinshu-u.ac.jp/ICSEA/ show the traceability link recovery result with f-measure. F-measure for asymmetrical similarity is better than symmetrical similarity for 14 data, but worse for only one data, so in some cases, asymmetrical similarity is better method.

From experiment 1, the necessity of the adequate method instance selection and the effectiveness of asymmetrical similarity become clear.

B. Experiment 2: Cross Validation

Experiment 2 is the traceability link recovery method mining experiment. First we try cross validation in order to evaluate the possibility of our proposed method. We integrated the results of the experiment 1 into one data (weka format file) and did 10 fold cross validation by using several algorithms.

TABLE III
THE IDENTIFICATION PERFORMANCE (CROSS VALIDATION)

	algorithm	p=0.5 r=0.7	p=0.3 r=0.7	p=0.2 r=0.6	p=0.1 r=0.5
precision	J48	0.92	0.87	0.89	0.96
precision	Naive Bayes	0.04	0.08	0.15	0.37
precision	Logistic	0.62	0.58	0.55	0.64
precision	Random Forest	0.72	0.75	0.80	0.95
recall	J48	0.54	0.68	0.82	0.99
recall	Naive Bayes	0.69	0.78	0.75	0.96
recall	Logistic	0.09	0.06	0.08	0.46
recall	Random Forest	0.54	0.62	0.74	0.95

TABLE IV
THE IDENTIFICATION PERFORMANCE (BY PROJECT VALIDATION)

precision	recall	# of data in which the traceability link recovery performance satisfies the criterion	# of projects whose identification precision is greater than 0.7	# of projects whose identification precision is greater than 0.5
0.7	0.7	4	0	0
0.6	0.7	20	0	0
0.5	0.7	87	1	1
0.4	0.7	164	0	1
0.3	0.7	358	1	1
0.3	0.6	655	2	4
0.2	0.7	1195	2	5
0.2	0.6	1873	3	4
0.1	0.5	9807	21	26
0.1	0.4	11238	23	27
0.05	0.5	16985	26	31

We show the identification performance for several traceability link criteria in Table III. Precision and recall in the first column mean the precision and recall of identification performance, and p and r in the first row mean the precision and recall of traceability link recovery performance.

We got acceptable precision values for each traceability link criterion, so the potential of our method becomes clear, but the dependency on algorithms is very high. Even for the objective traceability link criterion ($precision > 0.3$ and $recall > 0.7$), the precision is 0.87 in J48 and 0.75 in Random Forest, but 0.08 in Naive Bayes. These experiments only show the average performance, and in order to show the possibility for each data, the next experiment is needed.

C. Experiment 3: Identification performance check for each data

Our training data contains 1056 data for each project, that is, 36960 data. In cross validation these 36960 data is divided into ten subsets, and the combination of nine subsets is used as training data and the remaining one subset is used as test data, so the training data includes many data whose project is the same as the data in the test data. In the objective traceability link criterion, the number of candidate method instances is 358, that is only 1% (358/36960), and the number of projects which have candidate method instance is 10 (total is 35), so the bias between training data and test data may exist. As the result of this condition, identification performance may be overestimated, so we evaluate the performance for each project.

We constructed training data from N-1 data, and test data from the remaining data. In our experiment, N=35, so we constructed 35 pairs. We show the summarized result in Table IV and the detailed result in http://cwww.cs.shinshu-u.ac.jp/ICSEA/exp3.xlsx. 11 kinds of traceability link criterion are tested and the results are shown in Table IV.

For the objective traceability link criteria ($precision > 0.3$ and $recall > 0.7$), identification performances are low, that is, only one data satisfies the identification criterion ($precision > 0.7$). In almost all data, precision is zero even for very low traceability link criteria. We can get only the reasonable value, that is, identification performance ($precision > 0.7$) and traceability link recovery performance ($precision > 0.3$ and $recall > 0.7$) in the case of EasyClinic ID_CC, and traceability link recovery performance ($precision > 0.3$ and $recall > 0.6$) in the case of EasyClinic UC_ID.

This result can not show the potential of our method, so we did further experiments in order to consider the reasons and the possibility to improve identification performance.

We considered the following reasons:

- The number of candidate method instances is too small. As shown in Table I, the number of candidate method instances is too small compared with the number of tested method instances (1056) and deviation is large. In the case of $precision > 0.3$ and $recall > 0.7$, 11 out of 35 projects have the value zero, and further the percentage of the candidate method instances are low, that is, the most high case is for EasyClinic ID_CC and the value is 8% (89/1056). Training of succeeful pair based on a few succeeded data is very difficult, so better traceability link recovery methods or customization are needed in order to augument the number of candidate method instances.
- Each data has special link characteristics. For example, in SMOS the average number of link is especially larger than others, and in WV_CCHIT the number of destination components is larger than others, so the adequacy of method instance is a little different from each other and as the result of this difference the training data generated from such inadequate data becomes inadequate.

  We did method mining experiments for SMOS and WC_CCHIT by using selected training data in order to evaluate the matching of a training data and a test data. The detailed results are shown in http://cwww.cs.shinshu-u.ac.jp/ICSEA/ and summarized result is in Table V, which shows only the top two results. “all” in training data means the original experiment 3. # of T means the number of candidate method instances. The first 4 rows show that the identification performance is 0.145 for the case of using all data, but is 0.687 (0.466) for the case of using EAnci_UC_CC (Waterloo grp09). This result shows that the adequate training data improves the identification performance, that is, the identification performance by using adequate training data is larger than the case of the integrated training data. This adequacy may depend on the data characteristics, but the identification of these characteristics is now an open problem.

V. Threats to Validity

Regarding the internal validity, the variation of alternatives and the characteristics of documents are not sufficient. This research is still ongoing, and the main objective of this paper is to show the potential of the proposed method, so the result is not sufficient. The following method options and characteristics are to be considered:

- Further method options
  - Latent Dirichlet Allocation(LDA) and/or ontology application [18]
  - Granularity factor (How to divide a document into components)
  - What kind of term is to be used
- Document (pair) characteristics
  - Variance of similarity
  - Refined classification of document pair (in Easy-Clinic data set, there are four kinds of documents)
  - Language information ([7] shows that the English version and the Italian version have different result)
  - Development member variation
  - Estimated value of the number of traceability links

Regarding the external validity, the used data sets are not sufficient. We did not test all of the CoEST data set. Also real software documents have many variation (plain text, office document, CAD based document, etc) and have language problems. Granularity of components is the important factor, but CoEST data set is already separated as link unit, so other granularity is not tested. Our result only shows the availability of proposed method ming, so in order to apply to some real softwares, the corresponding knowledge base needs to be constructed.

VI. Related Research

There are many researches about traceability recovery and there are several methods categories: rule base, IR base, and format base. IR based method has wide availability because it entails no constraint to developers, but as the result of this weak constraint, accuracy is not so good. For IR based method, in order to improve accuracy, several methods are proposed and evaluated [4]–[12], [19]. Lucia [8] evaluated the effect of term identification methods. Wiese [10] considered the stemming effect, and Mahmoud [11] considered how to construct a term-document matrix. Capobianco [6], [7] and Lormans [20] compared several IR based traceability recovery methods: Jenson-Shannon Method(JS), Vector Space Model(VSM), LSI, LDA. Lormans said that the adequacy of these methods are dependent on the kind of document.

Several criteria are used for traceability link candidate judgment. Lormans [20] compares the following five methods and concludes that the adequacy is dependent on the kind of document:

- Cut point: we select top k links with similarity value
- Cut percentage: we select k percentage of the ranked list
- Constant threshold: we select those links that have a similarity measure greater than k
- Variable threshold: we select those links that have a similarity measure greater than k, where k is calculated according to a percentage of the total similarity measures
- Scale threshold: we select links according to k = c * MaxSimilarity where $0 \leq c \leq 1$

There are many researches which evaluate several methods, but external validity is not considered sufficiently, so engineers cannot select/use the adequate method for their projects.

VII. Conclusion and Future Work

We proposed traceability link recovery method mining in order to select adequate method instances and to assure the traceability link criterion. Our experimental result shows the potential of our proposed method, but the accomplished identification performance is not sufficient. Regarding the research questions (RQ1 and RQ3), the answer is yes for some projects, but no for the other projects. It shows the heavy project dependency. In order to resolve this dependency, we need to improve both traceability link recovery performance and identification performance. For the former improvement, the following alternatives are planned:

1) LDA and statistic model
2) Candidate link judgment. Lormans [20] defined five judgment methods. We only used two, so the remaining three methods are to be evaluated.
3) There may be several categories about the document link properties, so the similarity functions which are adequate for these links are needed.

For the latter improvement, the followings are planned;

1) There exist several reference data, which are not evaluated, so we do further experiments using those data and consider the matching of training data and test data.
2) The used characteristics are not sufficient, so we consider the characteristics which are more related with document link properties.

Regarding the research question (RQ2), the answer is almost yes, but the relation between the effectiveness and the document characteristics is not clear. Further consideration about this relation is needed.

References

[1] C. Wohlin, P. Runeson, M. Hoest, M. C. Chisson, B. Reqnell, and A. Wessln, Experimentation in Software Engineering. Springer, 2012.

[2] Z. He, F. Shu, Y. Yang, M. Li, and Q. Wang, “An investigation on the feasibility of cross-project defect prediction,” Automated Software Engineering, vol. 19, no. 2, 2012, pp. 167–199.

TABLE V
THE IDENTIFICATION PERFORMANCE FOR SMOS AND WV_CCHIT

test data	precision	recall	# of T	training data	precision	recall	# of T	identification performance (precision)
SMOS	0.05	0.3	137	all	-	-	-	0.145
SMOS	0.05	0.3	137	EAnci_UC_CC	0.05	0.4	187	0.687
SMOS	0.05	0.3	137	Waterloo grp09	0.1	0.5	164	0.466
SMOS	0.05	0.2	279	all	-	-	-	0.506
SMOS	0.05	0.2	279	EAnci_UC_CC	0.05	0.3	296	0.781
SMOS	0.05	0.2	279	EasyClinic CC_TC	0.1	0.5	262	0.665
WV_CCHIT	0.05	0.3	136	all	-	-	-	0.111
WV_CCHIT	0.05	0.3	136	EasyClinic ID_CC	0.2	0.7	164	0.402
WV_CCHIT	0.05	0.3	136	waterloo grp23	0.2	0.6	110	0.359
WV_CCHIT	0.05	0.2	258	all	-	-	-	0.376
WV_CCHIT	0.05	0.2	258	EasyClinic CC_TC	0.1	0.5	262	0.607
WV_CCHIT	0.05	0.2	258	EasyClinic ID_CC	0.2	0.6	224	0.550

[3] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor, “Software traceability with topic modeling,” in ICSE, ACM, 2010, pp. 95–104.

[4] A. Abadi, M. Nisenson, and Y. Simionovici, “A traceability technique for specifications,” in ICPC, 2008, pp. 103–112.

[5] M. Borg, K. Wnuk, and D. Pfahl, “Industrial comparability of student artifacts in traceability recovery research,” in CSMR, 2012, pp. 181–190.

[6] G. Capobianco, A. D. Lucia, R. Oliveto, A. Panichella, and S. Panichella, “On the role of the nouns in ir-based traceability recovery,” in ICPC, 2009, pp. 148–157.

[7] G. Capobianco, A. D. Lucia, R. Oliveto, A. Panichella, and S. Panichella, “Traceability recovery using numerical analysis,” in WCRE, 2009, pp. 195–204.

[8] A. D. Lucia, M. D. Penta, and R. Oliveto, “Improving source code lexicon,” IEEE Tr. on S.E., vol. 37, no. 2, 2011, pp. 205–227.

[9] A. D. Lucia, F. Fasano, R. Oliveto, and G. Tortora, “Recovering traceability links in software artifact management systems using information retrieval methods,” TOSEM, vol. 16, no. 4, 2007, pp. 1–50.

[10] A. Wiese, V. Ho, and E. Hill, “A comparison of stemmers on source code identifiers for software search,” in ICSM, 2011, pp. 496–499.

[11] A. Mahmoud and N. Niu, “Source code indexing for automated tracing,” in TEFSE, 2011, pp. 3–9.

[12] R. Oliveto, M. Gethersy, D. Poshyvanyky, and A. D. Lucia, “On the equivalence of information retrieval methods for automated traceability link recovery,” in ICPC, 2011, pp. 68–71.

[13] “Center of excellence for software traceability.” http://www.coest.org/. 08.01.2013.

[14] B. Ramesh and M. Jarke, “Toward reference models for requirements traceability,” IEEE Tr. on S.E, vol. 27, no. 1, 2001, pp. 58–93.

[15] W. Jirapanthong and A. Zisman, “Supporting product line development through traceability,” in APSEC, 2005, pp. 1–9.

[16] J. H. Hayes, A. Dekhtyar, and S. K. Sundaram, “Advancing candidate link generation for requirements tracing: The study of methods,” IEEE Tr. on S.E., vol. 32, no. 1, 2006, pp. 4–19.

[17] I. H. Witten and E. Frank, Data Mining - Practical Machine Learning Tools and Techniques. Morgan Kaufman, 2005.

[18] Y. Zhang, R. Witte, J. Rilling, and V. Haarslev, “An ontological approach for the semantic recovery of traceability links between software artifacts,” Software, IET, vol. 2, no. 3, 2008, pp. 185–203.

[19] A. Marcus, “Recovery of traceability links between software documentation and source code,” International Journal of Software Engineering and Knowledge Engineering, vol. 15, no. 5, 2005, pp. 811–836.

[20] M. Lormans and A. van Deursen, “Can lsi help reconstructing requirements traceability in design and test?,” in CSMR, IEEE, 2006, pp. 1–10.