2021 20th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

# Traceability method between design documents and source codes based on SQL dependency

Lujing Yu,Yonghua Li
*School of Computer Science and Technology,*
*Wuhan University of Technology*
*Wuhan, China*
*Email: 1591891342@qq.com, liyonghua@whut.edu.cn*

Yuqing Feng
*Academic Affairs Office,*
*Wuhan Social Work Polytechnic*
*Wuhan, China*
*Email: 280064139@qq.com*

Chen Qi
*Information Technology Department,*
*Chongqing Guoyuan Port Co, Ltd*
*Chongqing, China*
*Email: 358793710@qq.com*

***Abstract*—Information Retrieval (IR) technology was widely used in traceability between design documents and source codes. However, the vocabulary mismatch between the design documents and the source codes affects the performance of IR. Aiming at the above situation, a dynamic tracing method from design documents to source codes combining IR technology and SQL statement is proposed in management information system. Firstly, the similarity of the two is calculated by IR and the candidate links are generated; Then, the SQL statement required by the codes is automatically estimated according to the design documents, and the SQL statement is compared with the actual SQL statement in the codes to correct the design documents-codes similarity score; Finally, set a threshold to determine the trace links of the design documents to the source codes. The experimental results show that this method can improve the similarity score of code classes with relevant SQL statements in the design documents, so as to improve the ranking of code classes in the candidate links, extract the trace links that may be missing in IR method under the action of threshold, and finally improve the precision of trace results.**

***Keywords*-software traceability; SQL dependency; dynamic traceability; information retrieval; software engineering;**

## I. INTRODUCTION

Software Traceability refers to the ability to interrelate any uniquely identifiable software engineering artifact to any other, maintain required links over time, and use the resulting network to answer questions of both the software product and its development process[1]. There is a correlation between documents and codes. Discovering and maintaining the tracking links between design documents and codes helps software engineering activities such as program understanding, software maintenance, and requirements tracking.

At present, the most widely used software tracking method is Information Retrieval (IR), that is, by constructing IR model and identifying the tracking links according to the text similarity between software artifacts. Literature[2] first used IR technology to generate the tracking links of software documents and program codes. On the theoretical basis of early work[3], literature[4] proposes a concept of code dependency, called Closeness, to quantify the degree of interaction between classes based on code dependency and data dependency. Literature[5] extends the previous work combining direct code dependency and IR technology[6] through User Feedback. Literature[7] proposed a demand tracking method combining IR technology and non-text technology.

Most of the above researches firstly calculate the similarity from source codes to target file based on IR similarity model, establish the list of candidate links, then combine with other technologies (such as code dependency) to weight the similarity of candidate links, and finally set the threshold get the final tracking link. However, due to the different vocabulary usage between the documents and the codes, it is difficult to match, so these methods based purely on IR model are usually less accurate in establishing the documents-codes trace link.

This article is aimed at information management systems. In such systems, there are usually close database interactions. There are bound to be database operation statements in the design documents. These statements will be related to the SQL statements in the source codes[8]. Based on the description information in the design document, this paper predicts to generate the corresponding SQL statements, analyzes the correlation between the predicted SQL statements and the SQL statements in the actual codes, and optimizes the trace links generated by IR technology according to the analysis results.

## II. TRACING METHOD BASED ON SQL DEOENDENCY

The process of designing documents to codes tracing method based on SQL dependency is shown in Figure 1. It consists of five processing stages: first, the material is pretreated to obtain the domain term material; Then IR model is used to generate candidate links list between source material and code classes, generate source material (estimated SQL statements) evaluation unit token sequence, and generate actual SQL statements evaluation unit token sequence. Finally, the SQL dependency closeness between the source material evaluation unit and the actual SQL evaluation unit is calculated and analyzed.

### *A. Material preprocessing*

This paper extracts the title and function description from the design document as the source material docu-

2473-3636/21/$31.00 ©2021 IEEE
DOI 10.1109/DCABES52998.2021.00043

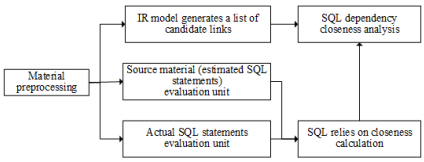


Figure 1. processing flow based on the SQL dependency tracking method.

ment; from the source code document, extracts the method comment, method body, and method namespace as the target material document. The actual SQL statements are also extracted from the source code document. Finally, this step extracts the database table information and the relationship between the tables from the database physical data model (PDM) document, as the domain term material.

### *B. IR model generates a list of candidate links*

This paper uses the Vector Space Model (VSM) method[2] to generate a list of candidate links between the source material and the code classes, including the following 4 steps: 1) create a Corpus; 2) Normalized the corpus; 3) Make corpus and calculate the text similarity; 4) Generate a list of candidate links

### *C. Cgeneration of evaluation unit sequence of source material(estimated SQL statement)*

The generation of source material evaluation unit sequence needs to parse the PDM document and the establish Database Table Information Struct (DTIS), and then extract the domain-related terms from the design document to establish a description term mapping set, which is used to map SQL operation type and SQL Predicate, etc. Before the conversion, Stanford Parser[9] is used to analyze the source material to determine the phrase corresponding to the phrase in the sentence and the part of speech of the word.

### *D. Generation of actual SQL statement evaluation unit sequence*

The accuracy of the trace links can be seriously affected by the possible differences in variable names and structures between actual and estimated SQL statements. Therefore, before the automatic tracing, the actual SQL statement needs to be normalized while ensuring the semantics. The conversion process of the evaluation unit sequence includes two steps: normalization of SQL sentences and lexical analysis.

### *E. SQL dependency and SQL dependency closeness calculation method*

*1) SQL dependency:* This paper refers to the relationship between the estimated SQL statements of the source material and the actual SQL statements as SQL dependency. The estimated SQL statements mentioned in this article refer to the SQL object converted from the source material combined with DTIS and the description term mapping set, for example, SQL keywords such as SELECT,WHERE, and database table or field names.

*2) SQL dependency closeness calculation method:* SQL Dependency closeness: Measure the correlation between the SQL object converted from the source material and the dependency of actual SQL statements.

a)Search item setting

This article compares the estimated SQL statements with the actual SQL statements from the following six aspects:

- F_keyword: Refers to keywords used to describe the structure of SQL.
- F_table: SQL statement operation master data table.
- S_table: F_table is a table related by a foreign key, also called a slave table.
- F_field: Represents the operation field.
- O_field :Represents the condition field.
- O_keyword: Other keywords (such as FROM, WHERE, etc).

Assuming that the weight of each unit type is $W_i$ ($i$ = 1-6) and the weight ratio of SQL at different levels is $(1/2)^{level-1}$, the token sequence of estimated SQL statements is $X$, and the token sequence of actual SQL statement is $Y$. The similarity calculation formula of sequence $X$ and $Y$ is shown in equation (1) :

$$sim(X,Y) = \begin{cases} \sum_{i=1}^{level-1} \frac{1}{2} \sum_{j=1}^{6} \frac{n_j}{N_j} \times W_j \\ 0, The\ F\_keyword\ of\ X \neq \\ \quad F\_keyword\ of\ Y \end{cases} \quad (1)$$

Where, $n_j$ represents the number of matches of each unit type in the Token sequence of $X$ and $Y$, $N_j$ represents the number of each unit type in $Y$, $W_i$ represents the corresponding weight, and $level$ represents the number of levels of SQL statements.

b) The steps to calculate the similarity of SQL statements

step 1: Analyze the source material and convert it into SQL object to form evaluation unit sequence (token sequence) $X$.

step 2: Tokenize the actual SQL statements in the source program, divide the evaluation units according to the order of the SQL statements and the separator specified by the syntax to form a token sequence $Y$, and count the number of types of units in $Y$ is $N_j$.

step 3: Count the matching number $n_j$ of each unit type between $X$ and $Y$ sequence at the same level; at the same time, calculate the similarity of SQL at a certain level by equation (1).

step 4: If the level of the actual SQL statement is $level$¿1, repeat step 2 to obtain the overall similarity of the $X$ and $Y$ sequences.

### *F. SQL dependency closeness analysis*

This section will introduce the IR value weighting algorithm between the source material and the code classes, so as to improve the IR value of candidate links generated in Section 1.2 and realize the reordering of the candidate link list. The weighted algorithm can not only weight the IR value of the candidate links of the class in which the actual SQL statement is located, but also get corresponding

weights for other classes that have a call relationship with the class.

**Definition1** : The token sequence and the weight of each retrieval item are:

$$W_{token} = (W_{F\_keyword}, W_{F\_table}, W_{S\_table}, W_{F\_field}, W_{O\_field}, W_{O\_field})$$

The calculation equation of the score points is shown in equation (2).

$$W_{token} = \begin{cases} W_{F\_keyword} = 0 \\ W_{F\_table} = W_{init} \\ W_{S\_table} = 1/2 W_{W_F\_table} \\ W_{field} = 1/4 W_{F\_table} \\ W_{O\_field} = 1/20 W_{F\_table} \\ W_{O\_keyword} = 1/20 W_{F\_table} \end{cases} \quad (2)$$

Through the test of the data set, it is found that when $W_{init}$=0.2, the overall effect of improving the candidate links list based on SQL dependency closeness is the best, so 0.2 is selected as $W_{init}$.

**Definition2** : Suppose the target class set of the candidate links list is $C_T$, the target class $C_{Ti}$ calls the $C_{DAL}$-like method set in the DAL layer as $M$, and the weight that $C_{Ti}$ can obtain based on the SQL dependency is $W_{SQL}$. The weight of $C_{DAL}$ is $W_{DAL}$.

The calculation equations of $W_{SQL}$ and $W_{SQL}$ are shown in equation (3) and equation (4):

$$W_{SQL} = \frac{1}{n}(\sum_{i=0}^{N} sim(X, Y_i)) \quad (3)$$

$$W_{DAL} = Bonus_{max} = Max(W_{SQLi}), i = 0, 1, ..., |C_T| \quad (4)$$

Among them, $N$ is the number of methods $|M|$, and $Y_i$ represents the token sequence of the SQL statements in the method $m_i (m_i \in M)$ . Based on the SQL dependency closeness analysis, weight the IR values of $C_{SQL}$ and $C_{DAL}$.

## III. Experiments and analysis

### A. Experimental data set

The experimental materials in this paper are from the design documents (source material) and sources code (target material) of a port production business management system (PCTOS) in Chongqing. The PCTOS includes four subsystems: business, dispatch, tally, and materials. Table 1 records the source material and target material types, the number of materials, and the number of correct tracking links that need to establish a tracking link relationship in the PCTOS system. In this paper, Visual Studio 2015 was used to export the code diagram DGML file and obtain the calling relationship of the classes. At the same time, the PDM document of PCTOS is also used in the experiment to obtain the domain corpus.

### B. Analysis of results

For the setting of the threshold, this experiment first tests the material system data set, and uses the F2Measure metric to evaluate the accuracy of the tracking link returned on the 4 sets of data (averaged). F(0.1)

Table I
Experimental data set, materials and number of correct links

| Subsystem | Number of materials (pieces) | | Correct links (pieces) |
|---|---|---|---|
| | Source material | Source material | |
| Business | 32 | 132 | 96 |
| Dispatch | 30 | 120 | 92 |
| Tally | 26 | 110 | 81 |
| Meterials | 32 | 140 | 101 |

represents the threshold $Thresholdlink$=0.1, similarly $Thresholdlink$=0.2, 0.3 corresponds to F(0.2), F(0.3). When Thresholdlink¿0.3, several comparison methods, tracking link recall is greatly reduced, while the accuracy is not significantly improved, so this paper selects 0.1, 0.2 and 0.3 to set the threshold.

From Table 2, the F(0.1) of the four sets of data based on the SQL dependency closeness analysis method is higher than F(0.2), F(0.3), so the threshold $Thresholdlink$=0.1 is the most appropriate; although the VSM method is in the first F(0.2) is the largest in the two sets of data (16 samples), but F(0.1) is the highest in the other sets of data, so $Thresholdlink$=0.1; in the same way in literature[7], threshold $Thresholdlink$=0.2.

The box plot generated by the R tool[10] is shown in Figure 5. The X axis represents the three methods compared in the article, and the Y axis represents the recall, precision, and F2Measure of the three methods. The box plot shows the minimum, the first quartile (the lower line of the box), the median (the line in the middle of the box), the third quartile (the upper line of the box), and the maximum from bottom to top. Any point that falls further will be considered an "extreme" value and will be drawn separately. It can be seen from the results in Figure 2 that the recall rate based on the SQL dependency closeness analysis method and the literature[7] method is not obvious compared to the VSM method, but the precision rate and F2Measure are better. In terms of precision and F2Measure indicators, the method in this paper is superior to the other two methods in terms of median and first quartile. This shows that the analysis method of combining IR technology and SQL dependency proposed in this paper is effective for the correction of the candidate links list generated by VSM.

## IV. Conclusion

This paper proposes a method of extracting trace link from design documents to source codes based on SQL dependency. The method first extracts the function description from the design document with chapter granularity as the source material, extracts key information such as method content and method namespace from the source code document with the method granularity as the target material, and uses the VSM model to calculate the similarity score between the source material and the target material, and then obtain the candidate links list of the source material and the code, sorted in reverse order by

Table II
F2MEASURE INDEX EVALUATION RESULTS

| Data Group | VSM | | | Literature[7] | | | Method of this article | | |
|---|---|---|---|---|---|---|---|---|---|
| | F(0,1) | F(0,2) | F(0,3) | F(0,1) | F(0,2) | F(0,3) | F(0,1) | F(0,2) | F(0,3) |
| 8 | 0.565 | 0.532 | 0.524 | 0.616 | 0.601 | 0.551 | 0.685 | 0.643 | 0.589 |
| 16 | 0.534 | 0.567 | 0.507 | 0.563 | 0.592 | 0.521 | 0.642 | 0.630 | 0.605 |
| 24 | 0.570 | 0.487 | 0.461 | 0.539 | 0.587 | 0.526 | 0.658 | 0.601 | 0.587 |
| 32 | 0.572 | 0.530 | 0.473 | 0.548 | 0.593 | 0.515 | 0.690 | 0.598 | 0.563 |

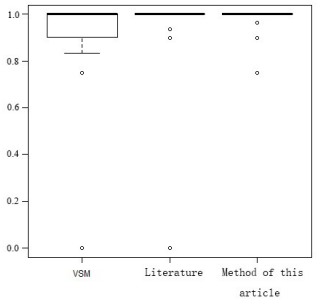


Figure 2. Recall

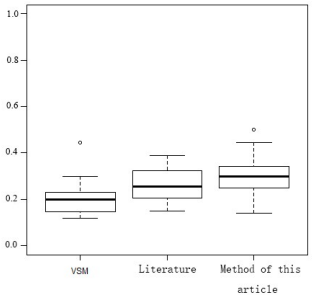


Figure 3. Precision

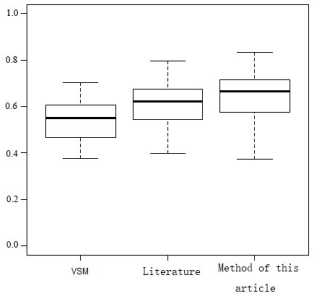


Figure 4. F2Measure

Figure 5. Box plot of experimental results.

the IR value; then combine the dependency relationship between the source material and the actual SQL statements in the code, calculate the SQL dependency closeness of the source material and the actual SQL statements, and then obtain the weight of the source material and the code category, and adjust the IR value in the source material and the code category candidate links; Finally, the threshold is set to determine the tracking link of the source material and the code category. For the same experimental object, this method improves the precision rate compared with the traditional IR method. However, when converting the source material into the evaluation unit sequence, only the single-level SQL structure is considered at this stage, and the nesting is not involved. Second, the set of rules used to eliminate ambiguity in the conversion process is limited, which can be continuously expanded in subsequent work. In the follow-up work, user feedback will also be combined with the method in this article to achieve better results.

## REFERENCES


[1] CoEST. Center of excellence for software traceability[EB/OL]. [2020-06-02]. http://www.CoEST.org.

[2] Salton G,Wong A,Yang C S. A vector space model for automatic indexing[J].Communications of the ACM,1975, 18(11): 613-620.

[3] Mona Rahimi ; Jane Cleland-Huan. Evolving Software Trace Links between Requirements and Source Code[J]. Empir. Softw. Eng, 2018,23(4):2198-2231.

[4] Hongyu Kuang, Hui Gao, et al.Using frugal user feedback with closeness analysis on code to improve IR-based traceability recovery[J]. ICPC 2019: 369-379.

[5] Panichella A, Mcmillan C, Moritz E, et al.When and How Using Structural Information to Improve IR-Based Traceability Recovery[J]. The 17th European Conference on Software Maintenance and Reengineering, 2013, 88(2):199-208.

[6] Shiheng Wang , Tong Li , Zhen Yang. Exploring Semantics of Software Artifacts to Improve Requirements Traceability Recovery: A Hybrid Approach[C]. Asia-Pacific Software Engineering Conference (APSEC),2019.

[7] JYOTI,CHHABRA J K.Requirements traceability through information retrieval using dynamic integration of structural and co-change coupling[C]// Proceedings of the International Conference on Advanced Informatics for Computing Research.Berlin German: Springer,2017: 107-118.

[8] Hang Li. Learning to Rank for Information Retrieval and Natural Language Processing: Second Edition [M]. Morgan & Claypool,2014.

[9] Stanford University. The Stanford Parser: A statistical parser[EB/OL] [2017-06-09]. https://nlp.stanford.edu/software/lex-parser.shtml

[10] Auckland University.R[EB/OL].(2020-06-20),[2020-07-30].https://www.r-project.org.