

SEMANTIC TOPOLOGICAL RELATION BASED USE CASE DIAGRAM RETRIEVAL

SHENGSHENG WANG, YANG LIU AND HAIYANG JIA*

College of Computer Science and Technology
Jilin University

No. 2699, Qianjin Street, Changchun 130012, P. R. China
wss@jlu.edu.cn; liu_yang14@mails.jlu.edu.cn; *Corresponding author: jiahy@jlu.edu.cn

Received December 2015; accepted March 2016

ABSTRACT. *It is of great significance to identify the semantic similarity of software engineering use case diagrams (UCDs) in degree thesis duplication checking. However, the existing image retrieval methods can only recognize UCDs that have similar appearances except those sharing similar semantics. To this end, we address a new scheme for UCD retrieval based on semantic topological relations (STRs). In our approach, a semantic topological relation graph (STRG) is used to describe the structural information of a UCD and the similarity matching work is completed by using graph matching algorithm. Experimental results demonstrate that the proposed method performs more favorably against other alternatives in terms of identifying the semantic similarity.*

Keywords: Content-based image retrieval, Use case diagram retrieval, Semantic topological relation, Graph matching, Degree thesis duplication checking

1. Introduction. There are a certain number of plagiarism phenomena in Chinese software engineering dissertation. Although China has a strong paper duplication checking system, it can only retrieve the text except images. In fact, there is a lower text repetition rate but a higher use case diagram (UCD) similarity degree in these plagiarized degree theses. There should be no similar UCDs in different software systems since they actually reflect users' personalized requirements. Although some existing methods can recognize UCDs sharing similar appearances, they can hardly work well when the UCDs have different appearances despite their same semantics, as is shown in Figure 1 and Figure 2.

There are many image retrieval approaches that can be applied to UCD retrieval, including document retrieval, circuit diagram symbol retrieval, engineering drawings retrieval and hand drawings retrieval, etc. Common documents retrieval methods are mainly divided into two types: character content-based retrieval [5] and image feature-based retrieval [3]. The former depends on OCR techniques while the latter is based on image matching. In [7], a new scheme based on BoRs is proposed for symbol retrieval. Most engineering drawings retrieval methods are based on text keywords, resulting in much time for annotation and lower retrieval efficiency. Recently, visual content (shape and spatial relations) [10] has been added into engineering drawings retrieval methods. The key of hand drawings retrieval lies in graph primitives fitting and recognition, and [6] presents a novel approach based on sketch keyshapes. However, after testing the above traditional image retrieval methods, we found that they are not quite suitable for UCD retrieval.

In this paper, a new scheme for UCD retrieval based on semantic topological relations (STRs) is proposed. We define a semantic topological relation graph (STRG) to describe UCDs and the similarity matching work is completed by an improved graph matching algorithm. The architecture of the proposed method is depicted as Figure 3.

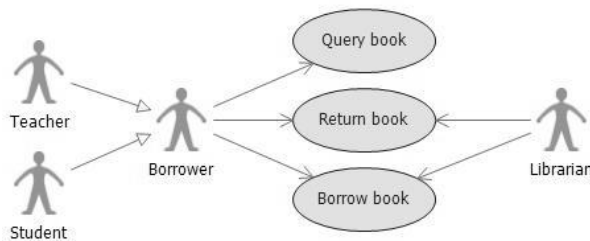


FIGURE 1. Use case diagram 1

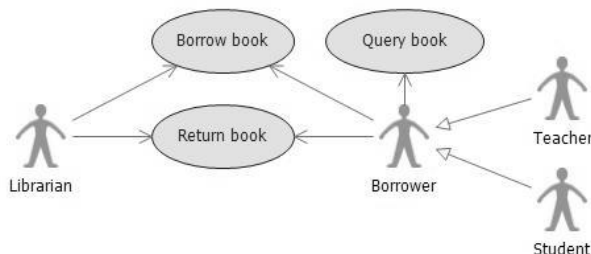


FIGURE 2. Use case diagram 2

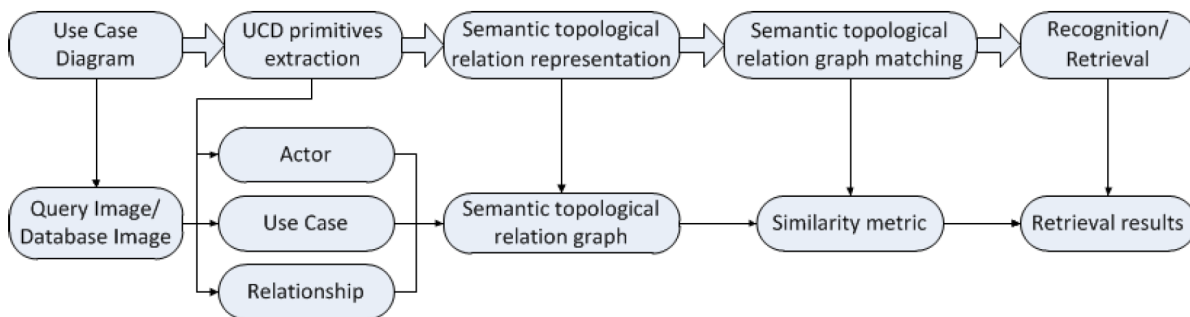


FIGURE 3. An architecture of the proposed method

The rest of this paper is organized as following. In Section 2, we introduce the STR-based approach. In Section 3, different experiments are performed to validate our method from different aspects. Finally, Section 4 draws a conclusion.

2. STR-Based Approach.

2.1. UCD primitives extraction. The UCDs in software engineering degree theses mainly contain the following elements: *actor*, *use case* and *relationship* [8].

1) *actor* primitive

Consider the *actor* to be a simple shape (a simple closed curve in the plane), then we employ an improved template matching algorithm.

2) *use case* primitive

We employ an ellipse extraction algorithm to detect the *use case* which is a standard elliptical shape in the UCD.

3) *relationship* primitive

We mainly extract straight lines [4], dashed lines and arrows/arrowheads [1].

2.2. Semantic topological relation representation. The topological relations between UCD primitives are a kind of semantic feature. In order to facilitate the topological relations similarity comparison, we use a semantic topological relation graph (STRG) to describe the semantic topological relations (STRs) in a UCD.

Definition 2.1. A semantic topological relation graph (STRG) corresponds to a four-tuple $G = (V, E, AV, AE)$, in which:

- 1) V is the set of nodes which correspond to actors or use cases in a UCD;
- 2) E is the set of edges which correspond to relationships in a UCD;
- 3) AV is the set of node attributes which can identify which UCD primitive each node refers to in a UCD;
- 4) AE is the set of edge attributes which can identify which relationship each edge refers to in a UCD.

In the STRG, each edge has a direction and a value, different values refer to different relationships and each edge's direction keeps the same as that of corresponding relationship in the UCD. The STRG of Figure 1 can be depicted as Figure 4.

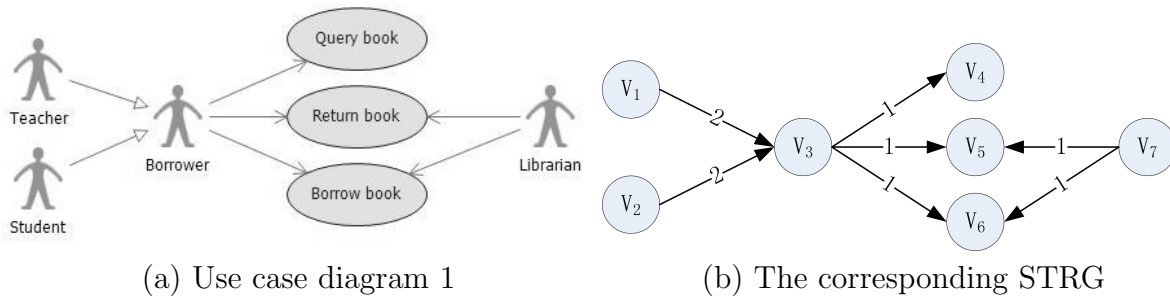


FIGURE 4. Examples illustrating the UCD and its corresponding STRG

2.3. Semantic topological relation graph matching. Based on our STRs, the second key work of UCD retrieval lies in similarity matching. Indeed, the STRG is essentially a weighted directed graph, so our algorithm will be divided into two steps in order to avoid the NP-hardness of the underlying graph matching problems.

We firstly take the STRGs as common directed graphs without considering their node attributes and edge attributes, and then a directed graph matching algorithm is performed on them to find their common paths. After that, we find the common edges in the common path by comparing their node attributes and edge attributes respectively.

The simplified directed graph matching algorithm [2] is described as follows:

Suppose A is the adjacency matrix of G , $P^n = A^1 + A^2 + \dots + A^n$, and then the total number of different length of paths in graph G is:

$$\sum_{i,j} P^\infty(i, j) = T_G \tag{1}$$

where $P^n(i, j)$ is the number of all the paths from node i to node j whose length is not more than n and $P^\infty(i, j)$ is the number of different length of paths from node i to node j .

Given two directed graphs, we can obtain their similarity degree by calculating the number of common paths with different length. More specifically, suppose $T_{G,G'}$ is the number of common paths in directed graphs G and G' , and their paths similarity degree is:

$$SimDigraph = \frac{T_{G,G'}}{\sqrt{T_G \cdot T_{G'}}} \tag{2}$$

For any two directed graphs G and G' , the number of their common paths can be calculated through the Boolean operations on their adjacency matrixes:

$$T_{G,G'} = \sum_{i,j} t_{G \oplus G'}^\infty(i, j) \tag{3}$$

Based on the above work, we can get a set of directed graph G' which is similar to directed graph G , suppose directed graph G is the corresponding STRG to the query UCD, and specific steps are described in Algorithm 1.

Algorithm 1: Common edges matching

Input: a set of directed graph G' which is similar to directed graph G ;

Output: the edge matching rate between G and G' ;

1. For each of G' , find the corresponding edges in G by comparing their edge attributes;
2. Compare the node attributes of the corresponding edges in G and G' ; if they correspond to the same node attributes, the number of common edges matching increases by 1;
3. If the ratio between the number of common edges matching and the total number of edges in G exceeds a certain threshold, we consider G and G' to be the same UCD.

Suppose the number of common edges matching is K and the total number of edges in G is $|E|$ and then the edge matching rate is defined as:

$$Match_rate = \frac{K}{|E|} \tag{4}$$

Thus, the similarity of G and G' can be obtained through the following calculation:

$$SimUCD = SimDigraph * Match_rate \tag{5}$$

3. Experiments. In this section, some relevant experiments are performed to evaluate the similarity matching ability of our method with respect to the other approaches, including BoR [7] and AHDH [9]. The dataset images in our experiments are selected from some related software engineering degree theses. There are 1000 UCDs in the dataset and they are divided into 25 categories. Some sample images are as follows.

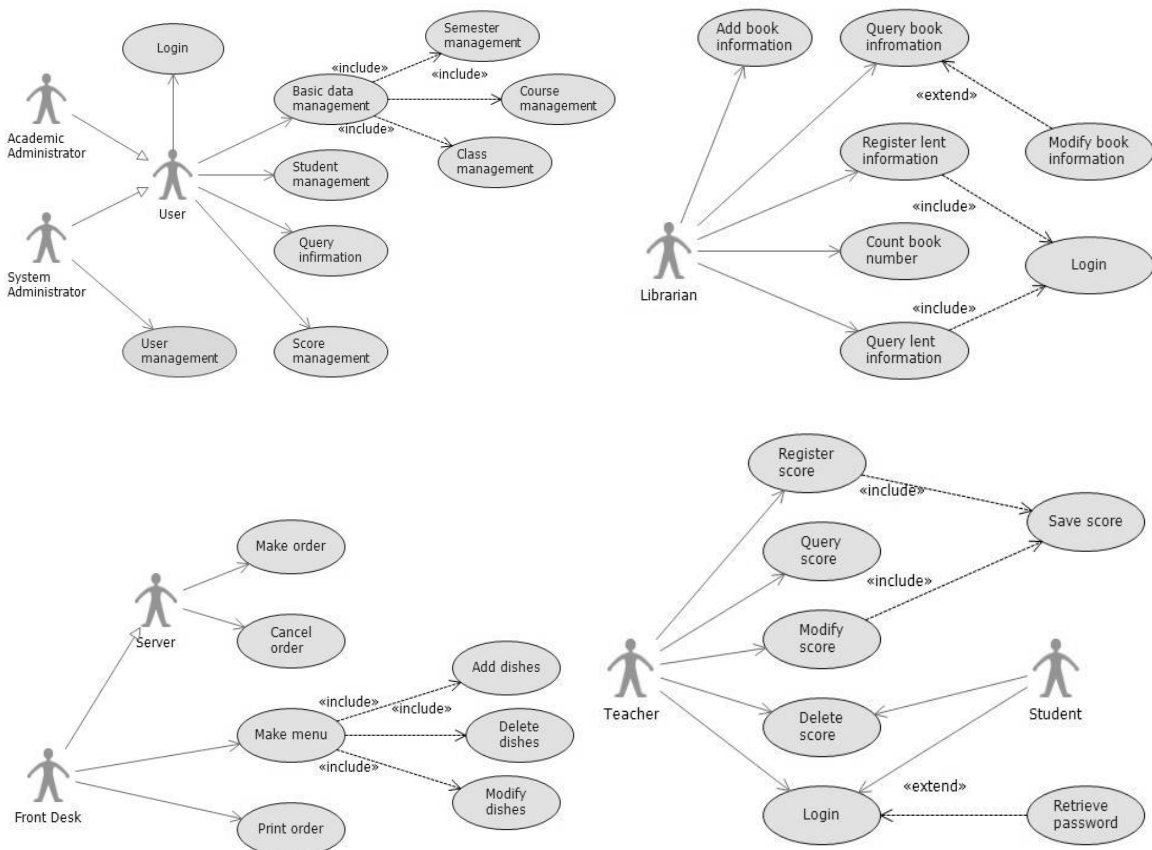
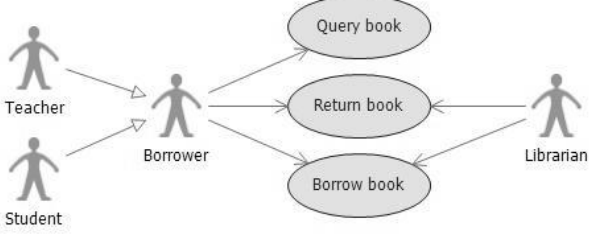
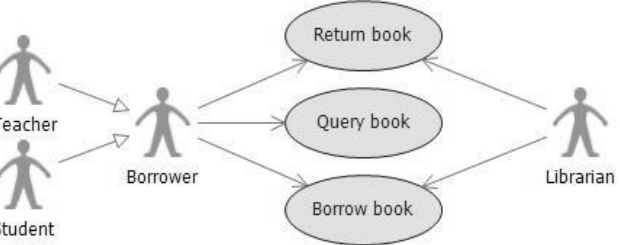
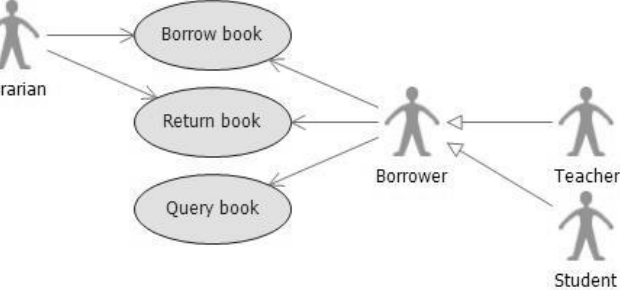
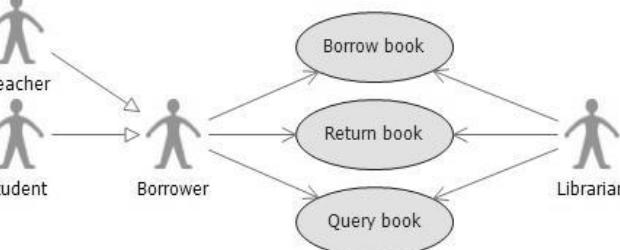
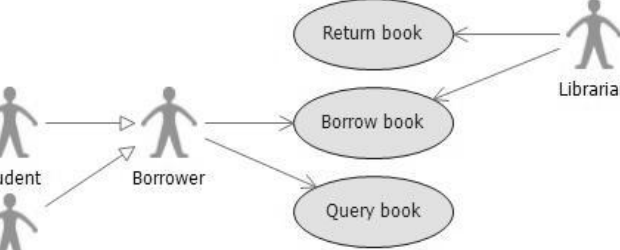


FIGURE 5. Sample images in the dataset

TABLE 1. The top 4 query results of the proposed method

<p>Query Image</p>	 <p>The diagram shows a central 'Borrower' node. To its left, 'Teacher' and 'Student' nodes have arrows pointing to 'Borrower'. To its right, 'Librarian' has arrows pointing to 'Borrower'. Three book nodes are positioned between them: 'Query book' at the top, 'Return book' in the middle, and 'Borrow book' at the bottom. Arrows connect 'Borrower' to each book node, and 'Librarian' to each book node.</p>
<p style="writing-mode: vertical-rl; transform: rotate(180deg);">Retrieval results</p>	<p>Rank1</p>  <p>The diagram shows a central 'Borrower' node. To its left, 'Teacher' and 'Student' nodes have arrows pointing to 'Borrower'. To its right, 'Librarian' has arrows pointing to 'Borrower'. Three book nodes are positioned between them: 'Return book' at the top, 'Query book' in the middle, and 'Borrow book' at the bottom. Arrows connect 'Borrower' to each book node, and 'Librarian' to each book node.</p>
	<p>Rank2</p>  <p>The diagram shows a central 'Borrower' node. To its left, 'Librarian' has arrows pointing to 'Borrower'. To its right, 'Teacher' and 'Student' have arrows pointing to 'Borrower'. Three book nodes are positioned between them: 'Borrow book' at the top, 'Return book' in the middle, and 'Query book' at the bottom. Arrows connect 'Borrower' to each book node, and 'Librarian' to each book node.</p>
	<p>Rank3</p>  <p>The diagram shows a central 'Borrower' node. To its left, 'Teacher' and 'Student' nodes have arrows pointing to 'Borrower'. To its right, 'Librarian' has arrows pointing to 'Borrower'. Three book nodes are positioned between them: 'Borrow book' at the top, 'Return book' in the middle, and 'Query book' at the bottom. Arrows connect 'Borrower' to each book node, and 'Librarian' to each book node.</p>
	<p>Rank4</p>  <p>The diagram shows a central 'Borrower' node. To its left, 'Student' and 'Teacher' nodes have arrows pointing to 'Borrower'. To its right, 'Librarian' has arrows pointing to 'Borrower'. Three book nodes are positioned between them: 'Return book' at the top, 'Borrow book' in the middle, and 'Query book' at the bottom. Arrows connect 'Borrower' to each book node, and 'Librarian' to each book node.</p>

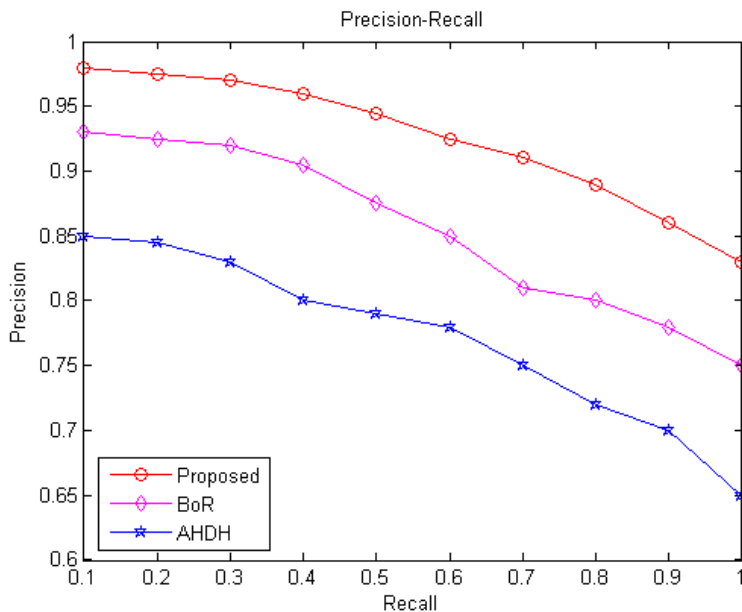


FIGURE 6. Precision-Recall graph comparing the proposed, BoR and AHDH methods

In our experiments, in order to qualitatively evaluate the retrieval efficiency of the proposed method, 4 most similar images corresponding to each query are retrieved from the dataset. Table 1 shows the top 4 UCDs in the ranked list with respect to the query image in Figure 1. We repeat this query 50 times, the experimental results present a good stability as well as efficiency of our method.

In order to quantitatively evaluate the retrieval efficiency of these methods, we use several commonly-used evaluation approaches in image retrieval, including precision, recall, and precision-recall. We randomly select images from the 25 categories, and for each category, 3 images are selected as query images to validate these methods. After 75 queries are performed, we can obtain the average precision-recall graph. In Figure 6, we show the precision-recall graph comparing the proposed, BoR and AHDH methods.

It can be seen from Figure 6 that the proposed method has achieved better performance compared with the others. This is probably because our method has taken the semantics in the UCDs into consideration based on our semantic topological relations (STRs) while none matters to semantics in the remaining approaches.

Finally, a further experiment is made by applying our method to the software engineering degree thesis duplication checking in Jilin University this year. We randomly select some UCDs in these dissertations as query images to test our method, and some UCDs that share similar semantics are retrieved from the dataset despite their different appearances, which cannot be recognized by BoR and AHDH approaches.

4. Conclusions and Future Work. In this paper, a new method for UCD retrieval based on semantic topological relations (STRs) is proposed. This method gives a unified representation to the structural information in UCDs. In this way, the problem of UCD retrieval is transformed into a graph matching problem. Experiments show that this method has achieved good performance. In the future, we will extend this work to other graphs in software engineering, then build a unified system for software engineering document retrieval.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (61472161, 61133011, 61402195, 61502198, 61303132, 61202308), Science & Technology Development Project of Jilin Province (20140101201JC).

REFERENCES

- [1] P. De, S. Mandal, A. Das and P. Bhowmick, A new approach to detect and classify graphic primitives in engineering drawings, *The 4th International Conference of Emerging Applications of Information Technology (EAIT'14)*, pp.243-248, 2014.
- [2] C. H. Elzinga and H. Wang, Kernels for acyclic digraphs, *Pattern Recognition Letters*, vol.33, no.16, pp.2239-2244, 2012.
- [3] R. Garg, E. Hassan and S. Chaudhury, Document indexing framework for retrieval of degraded document images, *The 13th International Conference on Document Analysis and Recognition (ICDAR'15)*, Tunisia, pp.1261-1265, 2015.
- [4] W. J. Kang, X. M. Ding, J. W. Cui and A. O. Lei, Fast straight-line extraction algorithm based on improved Hough transform, *Opto-Electronic Engineering*, vol.34, no.3, pp.105-108, 2007.
- [5] M. Kawamura, H. Kawanaka, T. Suzuki, H. Takase and S. Tsuruoka, A study on keyword detection using weighted similarity and character sequence for low-resolution medical documents, *The 17th International Conference on Informatics, Electronics and Vision (ICIEV'15)*, Singapore, pp.1-5, 2015.
- [6] J. Saavedra and B. Bustos, Sketch-based image retrieval using keyshapes, *Multimedia Tools and Applications*, vol.73, no.3, pp.2033-2062, 2014.
- [7] K. C. Santosh, L. Wendling and B. Lamiroy, BoR: Bag-of-relations for symbol retrieval, *International Journal of Pattern Recognition and Artificial Intelligence*, vol.28, no.6, 2014.
- [8] S. Sengupta and S. Bhattacharya, Formalization of UML use case diagram-a Z notation based approach, *2006 International Conference on Computing & Informatics (ICOCI'06)*, Malaysia, pp.1-6, 2006.
- [9] P. Sidiropoulos, S. Vrochidis and I. Kompatsiaris, Content-based binary image retrieval using the adaptive hierarchical density histogram, *Pattern Recognition*, vol.44, no.4, pp.739-750, 2011.
- [10] H. Song, P. Wang, X. Li and H. Li, Research and design of drawings retrieval system based on spatial relationships and shape, *ISECS International Colloquium on Computing, Communication, Control and Management (CCCM'08)*, pp.336-340, 2008.