Journal of Network Computing and Applications (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/jnca.2025.100108
ISSN 2371-9214 Vol. 10 Num. 1

Software requirement analysis and intelligent recommendation method combined with knowledge map

Ping Xu

School of Information Science and Technology, Taishan University, Tai'an, 271000, China

Keywords: Knowledge Map; Software Requirements Analysis; Intelligent Recommendation

Abstract: With the increasing complexity of software system, it is a key challenge to improve the accuracy of software requirements analysis and the effectiveness of intelligent recommendation. This paper focuses on exploring innovative methods of software requirements analysis and intelligent recommendation combined with knowledge map. By collecting multi-source information such as software requirements documents, user behavior data and feedback, the software requirements knowledge map is carefully constructed. On this basis, the software requirements are analyzed by graph mining technology and reasoning mechanism, and intelligent recommendation is realized by graph-based random walk algorithm with context information. In order to verify the effectiveness of the method, experiments were carried out with 100 enterprise management software users as samples. The results show that the accuracy of demand analysis in the experimental group is 85%, which is significantly improved compared with the control group's 68%. In terms of intelligent recommendation, the recall rate and accuracy rate of the experimental group were 86.7% and 87.5%, while those of the control group were only 61.1% and 64.3%. This shows that the method of combining knowledge map has achieved remarkable results in mining potential software requirements, understanding user requirements and providing accurate recommendations. It provides valuable new ideas for related research and practice in the field of software development, and is expected to promote the further innovation and development of technology in this field.

1. Introduction

In today's era of rapid digital development, the scale and complexity of software systems are constantly rising [1]. How to accurately analyze software requirements and provide users with efficient and intelligent recommendations has become a key problem to be solved urgently in the field of software development [2]. In this context, knowledge map has gradually emerged in this field with its powerful semantic processing and knowledge organization capabilities [3].

As the cornerstone of software development, software requirements analysis is directly related to the quality and practicability of software [4]. Traditional requirements analysis methods often face the dilemma of incomplete information acquisition and inaccurate understanding, and it is difficult to meet the increasingly complex and changeable business requirements [5]. Intelligent recommendation technology is widely used in many fields, but in the scenario of software requirements, how to combine software characteristics and user requirements to achieve accurate recommendation is still a research hotspot [6]. Knowledge map describes the relationship between entities in a structured form, which can effectively integrate the knowledge in the software field and provide rich semantic information support for demand analysis and intelligent recommendation.

This paper focuses on "software requirement analysis and intelligent recommendation method combined with knowledge map" to carry out in-depth research. The purpose of this paper is to explore how to use knowledge mapping technology to mine software requirements hidden in massive information and build an efficient intelligent recommendation model.

2. Intelligent recommendation system technology

Software requirements analysis aims to deeply understand users' expectations and requirements for software functions, performance and reliability, and turn them into detailed, accurate and verifiable requirements specifications [7]. Common methods include process-oriented analysis, describing system functions through data flow and data processing; Object-oriented analysis describes the system structure and behavior with concepts such as object, class and inheritance. However, these traditional methods are prone to problems such as information omission and understanding deviation when dealing with complex and changeable requirements.

Intelligent recommendation system uses algorithms to predict and recommend items that users may be interested in according to their historical behaviors, preferences and contextual information. Common technologies include content-based recommendation, which is recommended according to the matching degree between project characteristics and users' interest preferences; Collaborative filtering recommendation, recommending projects by analyzing the behaviors of similar users [8]. However, in the software requirement scenario, due to the professionalism and complexity of the software itself, it is difficult to directly apply the universal recommendation technology.

The essence of knowledge map is a semantic network, which stores and expresses knowledge in the real world in the form of triplets of entities, attributes and relationships [9]. Its construction process usually includes information extraction, extracting entities, attributes and relationships from various data sources; Knowledge fusion, eliminating conflicts and redundancies between different data sources; Knowledge processing, building ontology model and quality evaluation. Knowledge map can integrate the knowledge in software field, reveal the internal relationship between software entities, and provide a rich semantic information foundation for software requirements analysis and intelligent recommendation.

3. Software requirement analysis and intelligent recommendation method combined with knowledge map

3.1. Software requirements analysis method based on knowledge map

The powerful semantic representation and reasoning ability of knowledge map provides a new perspective for software requirements analysis. Firstly, the software-related entities, such as functional modules, user roles and business processes, are extracted from various requirements documents, user feedback and domain knowledge sources, and the relationships between entities, such as dependence, inclusion and operation, are determined. For example, in an e-commerce software demand analysis, there is a "dependence" relationship between the "commodity management module" and the "inventory management module" because the operation of putting goods on the shelves and removing them depends on the accurate record of the inventory quantity.

After constructing the knowledge map of software requirements, the requirements are analyzed by graph mining technology. Through path search, the potential relationship between different entities is found, which helps requirements engineers to mine hidden requirements. For example, if it is found that the "user complaint" entity is connected with the "payment function" entity through a series of relationships, it may imply that there is a potential problem with the payment function, which is an unspecified demand. In addition, based on the reasoning mechanism of knowledge map, new demand information is deduced according to the existing knowledge relationship. For example, it is known that the "new user" entity has the attribute of "unfamiliar with the operation process". Combining the relationship between "operation guidance function" and "user operation convenience", we can deduce the need to provide more perfect operation guidance function for new users.

3.2. Intelligent recommendation method combined with knowledge map

Intelligent recommendation in the scenario of software requirements needs to be closely combined with software knowledge and user-related information in the knowledge map. Firstly, user information, such as user behavior and historical needs, is associated with entities and relationships in the knowledge map. If a user frequently uses the "data analysis function" of a software, the relationship between the user and the "data analysis function" entity is established in the knowledge map. In the aspect of recommendation algorithm, the graph-based random walk algorithm is adopted. Starting from the entity node corresponding to the user, walk randomly in the knowledge map according to a certain probability. The nodes reached by each walk correspond to different software functions or demand options, and the walk probability is adjusted according to the strength of the relationship between nodes and user preferences.

Suppose that the knowledge map is a directed graph $G=(V,E)$, where V is a node set (corresponding to software functions, users and other entities) and E is an edge set (corresponding to the relationship between entities). The probability P_{ij} of transition from node i to node j can be defined as:

$$P_{ij} = \frac{\omega_{ij}}{\sum_{k \in N(i)} \omega_{ik}} \tag{1}$$

Where ω_{ij} represents the weight of the edge from node i to node j, which can be set according to the strength of the relationship between entities. $N(i)$ is the neighbor node set of node i, that is, the node set directly connected with node i. This formula ensures that the random walk is transferred according to the weight ratio of edges in the graph, which makes the walk more inclined to the nodes with close relationship, thus discovering other potentially valuable nodes related to the current node.

Consider the influence of contextual information on recommendation. Incorporate contextual information such as time and usage scenarios into the knowledge map to enrich the recommendation basis. In order to incorporate contextual information (such as time, usage scene, etc.), the weight ω_{ij} of the edge is dynamically adjusted. Assuming that the influence of context information C on the weight can be expressed by a function $f(C)$, the adjusted weight ω'_{ij} is:

$$\omega'_{ij} = \omega_{ij} \times f(C) \tag{2}$$

For example, in the enterprise quarter-end financial settlement scenario, the edge weight ω_{ij}

between function nodes related to financial statements is multiplied by a $f(C)$ value greater than 1 to improve the recommendation probability of functions related to financial statements in this scenario. In the daily business operation scenario, the edge weights of business process management related functions are similarly adjusted. In this way, the recommendation algorithm can more accurately recommend software functions that are in line with the current situation for users according to different contextual environments.

After many random walks, the access probability of nodes will gradually tend to a steady-state distribution π. By calculating the probability value of each node under the steady distribution, the priority of recommendation is determined. The probability π_i of the node i under the steady-state distribution satisfies the following formula:

$$\pi_j = \sum_{i \in V} \pi_i P_{ij} \tag{3}$$

And:

$$\sum_{i \in V} \pi_i = 1 \tag{4}$$

Finally, according to the π_{ij} value of each software function node from high to low, the top software functions are presented to users as recommended results. The higher the probability value, the stronger the correlation with the current needs of users under the influence of the structure and context information of the knowledge map, and the more it should be recommended first.

3.3. Algorithm model construction

Based on the above analysis and recommendation methods, an algorithm model combined with knowledge map is constructed. The model mainly includes three levels: data layer, knowledge map layer and application layer.

The data layer is responsible for collecting and preprocessing all kinds of data, including software requirements documents, user behavior data, context information and so on. After cleaning and labeling, these data provide the basis for the construction of knowledge map. The knowledge map layer uses the data processed by the data layer to construct the knowledge map of software requirements through technologies such as entity extraction and relationship mining. And use the management and maintenance tools of knowledge map to ensure the accuracy and consistency of knowledge. The application layer realizes the function of software requirement analysis and intelligent recommendation based on knowledge map. The proposed system identifies potential requirements through the requirements analysis module and delivers personalized software function or requirement recommendations to users via the intelligent recommendation module. During the operation of the model, the knowledge map is constantly updated and optimized according to the new data to improve the accuracy and timeliness of analysis and recommendation.

4. Experimental verification

The purpose of this experiment is to verify the effectiveness of software requirement analysis and intelligent recommendation method combined with knowledge map. The experiment takes enterprise management software as the research object, and selects 100 users with different experience to participate. Two main variables are set in the experiment: one is to use the method based on knowledge map to analyze and recommend requirements (experimental group), and the other is to use the traditional method (control group). The main evaluation indicators include the accuracy of demand analysis, the recall and accuracy of intelligent recommendation. The experimental data mainly come from users' actual operation records of enterprise management software, demand feedback questionnaires and software system logs. For the collected data, firstly, it is cleaned to remove duplicates, errors and incomplete data. Then label the data, for example, label the user's operation behavior corresponding to the software function module, and label the requirement feedback as functional requirement and performance requirement. The preprocessed data is used to construct knowledge map and compare the analysis and recommendation process of traditional methods.

The experimental group used the constructed knowledge map for software requirements analysis and intelligent recommendation, while the control group used traditional requirements analysis and recommendation technology. In the demand analysis stage, professional appraisers compare and evaluate the two groups of analysis results with the actual demand, and calculate the accuracy of demand analysis. The results are shown in Figure 1:

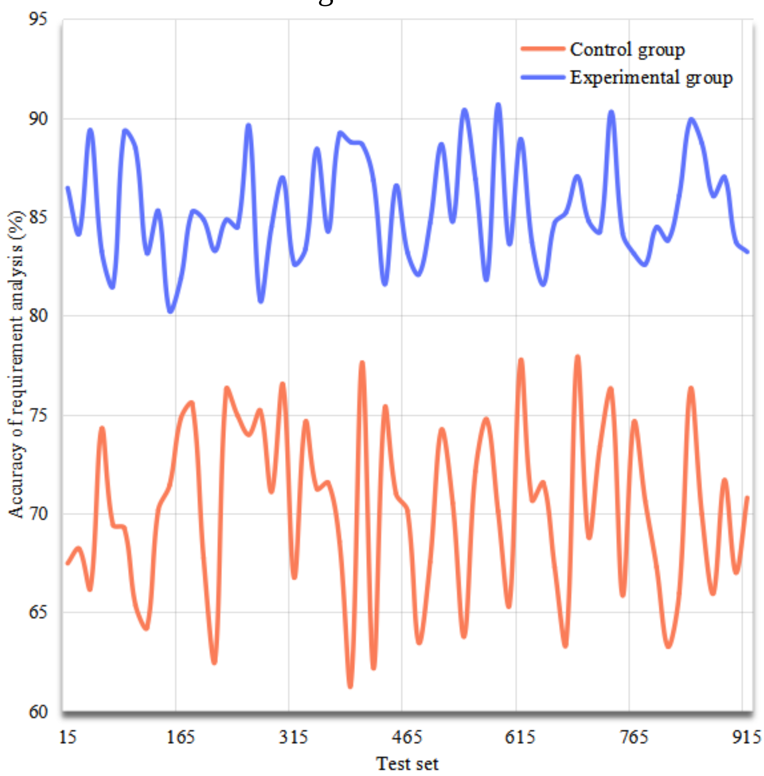

Figure 1 Accuracy of demand analysis

In the aspect of intelligent recommendation, users' feedback behaviors such as clicking and using the recommended content are recorded to calculate the recall rate and accuracy, as shown in Figure 2.

As can be seen from Figure 1, the accuracy of requirement analysis in the experimental group is obviously higher than that in the control group, which shows that the method based on knowledge map can mine software requirements more accurately. The powerful semantic understanding and reasoning ability of knowledge map is helpful to discover the requirement relationship hidden in complex data and reduce the requirement omission and misunderstanding.

Observing Figure 2, we can see that the experimental group also performed well in the recall and accuracy of intelligent recommendation. This is because the recommendation method combined with knowledge map can better understand the semantic association between user needs and software functions, make full use of contextual information, and provide users with more accurate and relevant recommendation content. However, traditional methods are relatively weak in dealing with complex semantic and contextual information. Combining the data of the two tables, the effectiveness and superiority of the software requirement analysis and intelligent recommendation method combined with knowledge map in practical application are verified.

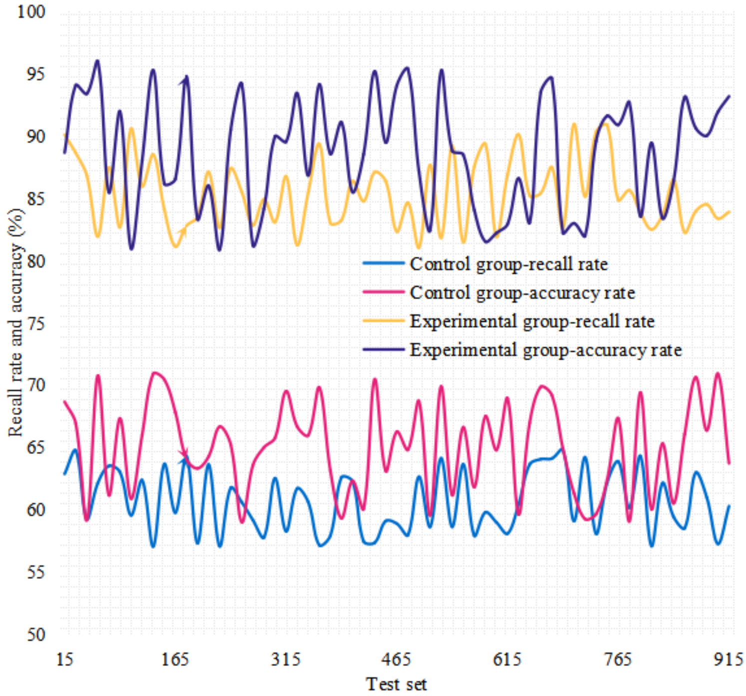

Figure 2 Comparison of recall and accuracy of intelligent recommendation

5. Conclusions

In this paper, the method of software requirement analysis and intelligent recommendation combined with knowledge map is deeply explored in order to meet the challenges brought by the increasingly complex software system. By constructing the knowledge map of software requirements, the multi-source heterogeneous data are integrated into a structured knowledge network, the in-depth analysis of software requirements is realized by graph mining and reasoning technology, and the intelligent recommendation is completed by using the random walk algorithm integrated with contextual information.

The experimental results strongly verify the effectiveness of this method. In terms of demand analysis, the accuracy of the experimental group is as high as 85%, while that of the control group is only 68%. This clearly shows that the method based on knowledge map can capture hidden requirements more keenly and greatly reduce omissions and misunderstandings in the analysis process. In the aspect of intelligent recommendation, the recall rate of the experimental group reached 86.7%, and the accuracy rate was 87.5%, which was in sharp contrast with the control group's 61.1% and 64.3%, highlighting the outstanding advantages of the new method in understanding users' needs and accurately recommending.

Nevertheless, there are still some limitations in this study. The construction of knowledge map is highly dependent on manual labeling data, which is not only costly, but also may affect the accuracy of the results due to subjective differences of the annotators. In addition, the experimental scenario is relatively limited, and the universality and expansibility of the method need to be further tested in more complex and diverse practical application scenarios. In the future, we can consider developing automatic and intelligent data labeling technology, and at the same time expand experimental scenarios to enhance the universality of the method. Generally speaking, the method of combining knowledge map brings a new dawn to the field of software requirements analysis and intelligent recommendation, which is expected to lead the technological change and innovation development in this field.

References

[1] Chazette L, Brunotte W, Speith T. Explainable software systems: from requirements analysis to system evaluation[J]. Requirements Engineering, 2022, 27(4):457-487.

[2] Kamal T, Zhang Q, Akbar M A. Toward successful agile requirements change management process in global software development: a client–vendor analysis[J]. IET Software, 2020, 14(3):265-274.

[3] Cui J, Cai Y, Wu Y. Criticality analysis for safety-critical software in nuclear power plant distributed control system[J]. Kerntechnik, 2021, 86(5): 343-352.

[4] Wang Shuo, Gu Yiran, Huang Liya. An Online Course Recommendation Model Based on Knowledge Subgraph and Attention Mechanism[J]. Computer Technology and Development, 2024, 34(4):139-145.

[5] Ruiz M, Hu J Y, Dalpiaz F. Why don't we trace? A study on the barriers to software traceability in practice[J]. Requirements Engineering, 2023, 28(4): 619-637.

[6] Yang Jiaqi, He Chaobo, Guan Quanlong, et al. A Relationship Prediction Method for Course Knowledge Graph Enhanced by Prerequisite Relationship Information[J]. Computer Science, 2024, 51(10):162-169.

[7] Habibullah K M, Heyn H M, Gay G, et al. Requirements and software engineering for automotive perception systems: an interview study[J]. Requirements Engineering, 2024, 29(1): 25-48.

[8] Zhang Yaqing, Shan Zhongyuan, Zhao Junfeng, et al. Knowledge Graph Instance Construction and Evolution Method Based on Intelligent Mapping Recommendation[J]. Computer Science, 2023, 50(6):142-150.

[9] Ducoffe M, Gabreau C, Ober I, et al. Certification of avionic software based on machine learning: the case for formal monotony analysis[J]. International Journal on Software Tools for Technology Transfer, 2024, 26(2): 189-205.