Available Online at http://www.recentscientific.com

International Journal of Recent Scientific Research

ISSN: 0976-3031

International Journal of Recent Scientific Research
Vol. 7, Issue, 12, pp. 14882-14886, December, 2016

Research Article

SOFTWARE REQUIREMENT TRACING AND REMOVING INCOMPLETE AND INCONSISTENT REQUIREMENTS AND PRIORITIZATION OF SOFTWARE REQUIREMENTS

Ranjith N*[1] and Marimuthu A[2]

[1]Department of Computer Science, Karpagam University, Karpagam Academy of Higher Education, Coimbatore, Tamilnadu, India
[2]Government Arts College, Coimbatore, Tamilnadu, India

ARTICLE INFO

Article History:

Received 10th September, 2016
Received in revised form 14th October, 2016
Accepted 08th November, 2016
Published online 28th December, 2016

Key Words:

Software engineering, software requirement, requirement tracing, semi-supervised ranking with graph based regularization, TD-IDF weighting schema.

ABSTRACT

Objectives: To trace software requirements and remove incomplete and inconsistent requirements of software and know the priority of requirements in a garment production management system. **Methods/Statistical analysis**: In multi-objective teacher-learning artificial bee colony optimization technique, optimal requirements are selected with the consideration of multiple constraints and multiple objectives. The main aim of this technique is to reduce cost and time consumption and maximize the client satisfaction that make use of constraints like cost threshold constraint, time threshold constraint and interaction constraints. Thus the selected requirements are traced for the purpose of removing the incomplete and inconsistent requirements. Requirement tracing is a process of requirement management that traces the requirements through their life cycle. The requirements are traced using TD-IDF weighting schema and firefly algorithm which are utilized to trace document requirement to requirements. From the traced requirements, there may be incomplete and inconsistent requirements that is removed by using logic based representation of requirements. It improves the quality of the software. Finally, complete requirements of the software is obtained and then prioritize the requirements based on case based ranking method to know which requirement is more important than the other. It follows three step process which are pair sampling, priority elicitation and priority learning. Priority learning is enhanced by semi- supervised ranking with graph based regularization where the semi-supervised labels are learned about the priority. **Findings:** From the selected requirements of software, trace those requirements from the document requirement to requirements and reject the requirements contain incompleteness and inconsistence and ranking the requirements to know the priorities of requirements. **Improvements/Applications**: The experimental results prove that the proposed techniques perform better than the existing technique in terms of precision-recall curve, threshold-f measure curve, precision and recall.

INTRODUCTION

Software engineering is an application which can be designing, implementing, developing, testing and maintaining a software. Software development is a field of software engineering in which software is developed to satisfy the customers. The software is developed based on the requirements which are collected from the customers, developers and users. The main objective is to attain customer satisfaction with tolerable resource limits. A major problem in the collection of requirements is incomplete, ambiguous, inconsistent or incorrect requirement due to improper communication amoung developers, customers and stakeholders. These issues lead to software hazards and it is curious that can be found out at the end stage of software development process. It can be overcome by tracing the life cycle of requirement and remove the incomplete, ambiguous and incorrect requirements from the list of requirements.

In an existing multi-objective teaching learning based optimization [Chaves-González, 2015] technique is used for selecting new set of requirements that can be included for the next release of the software. It considered the customer satisfaction and development cost of software to select the requirements. These selected requirements are traced using swarm technique and it can be enhanced by the proposed technique utilizing firefly technique [Elsood, 2014] for requirement tracing, it improves the performance of requirement tracing and it reduces false error rate. Then in the traced requirements incomplete, inconsistent requirements are removed by using logic based representation of requirements. After obtaining the complete requirements, the prioritization

**Corresponding author:* Ranjith N
Department of Computer Science, Karpagam University, Karpagam Academy of Higher Education, Coimbatore, Tamilnadu, India

[Saraç E, 2013] of requirements is given by case based ranking and presented semi-supervised ranking with graph-based regularization method for priority learning. It can be applied even for semi-supervised data. Hence, the complete requirements are obtained through the requirement tracing and found the priorities of requirement for the next software release.

Wang, Y., & Mao, W., proposed a list wise learning approach for ranking purpose called non linear list wise algorithm. This algorithm utilized clustering methods like K nearest neighbor method to find out single feature that describes the ranking order of hidden layers. This single feature developed weak rankers called Feature Rank and those are merged together through a boosting procedure.

Chen, S., *et al*. presented a ranking aggregation method called semi supervised ranking aggregation for the purpose of merging multiple ranking list. This ranking function is formed by merging the ranking list that is learned by the following rules. One of the rules is labeled and unlabeled items, the ranking score for labeled items are arranged in regular manner with the preference constraints. On the data diverse, the rank score of labeled items may vary and the weight of the ranking list is increased when a ranking list is similar to more ranking lists. The multiple ranking list is merged based on the hypothesis rules.

Sultanov, H., *et al*. presented a simple swarm technique and pheromone swarm technique for the requirement of engineering problem. The simple swarm technique and pheromone swarm technique are used to create the traceability matrices between the high level requirement and low level requirement which means the low level is traced to low level requirements. In this each particle trace the requirements independently and update their value based on global best and particle best. Thus optimal requirements are traced using swarm intelligence technique and pheromone swarm technique.
Wang, Y., *et al*. proposed an approach called Diff Rank for learning to rank in web search problem. The purpose of this approach is to create a new loss function in which the weak rank is transformed to strong rank and it also determines boosting objects in upper bound of training errors, then these upper bound are minimized simultaneously using the boosting process. By using Diff Rank approach obtains significantly the ranking model to define the ranking.

Cheng, X., *et al*. presented an approach called Manifold Ranking with Sink Points to define the problem of diversity in ranking mechanism. The critical problem of ranking is redundant objects, it may lead the improper top ranking. In this paper, diversity is taken into account for ranking. Manifold Ranking with Sink Points defines ranked points as sink points are objects assigned with minimum ranking score during the manifold ranking process on data manifolds thus it thwarts redundant objects obtained the high ranking score.

Perini, A., *et al*. defined a prioritization method for requirements named as case based ranking. In this method, machine learning approaches are used to evaluate the appended projects team member preferences with the software requirements ordering approximations, it gave more advantage for prioritization of requirements. The Rank Boost algorithm is utilized to prioritize the requirements based on case based ranking.
Sharma, R., & Biswas K. K., proposed an approach for incompleteness, ambiguity and inconsistency problem in software requirements called logical based representation of requirement. It analyzed the requirements and detected the incomplete requirements. It brought getter advantage during requirement engineering phase and it minimized the software hazards by reducing the number of defects at an early stage of software development.

Yigit, F., & Baykan, O. K., Presented particle swarm optimization and information gain technique for feature selection, classifier selection and feature categorization for text categorization. The proposed technique is used to choose the optimal features for text categorization. It improved the performance in terms of computation time of classifier by using optimal features alone for the text categorization process.

Zou, Y., *et al*. proposed an approach called interrogative-guided re-ranking approach for question oriented software text retrieval in software engineering. The interrogative-guided re-ranking generates the classifiers for six types of questions are whether, how to, which, why, where and what based on this the documents are determined with the answers for interrogative questions. These classifiers are combined together with open source text engine retrieval and the search results of text engine retrieval are re-ranked based on the classification score.

Goknil, A., *et al*. proposed an approach for automatic generation validation and creation of traces. This approach creates and validates the traces based on the architecture of software and requirements of software. The creation of traces involves two steps are architecture verification techniques and requirement relations. It utilized a tool for the purpose of defining a semantic relationship between requirements and traces and also it was used for validate those requirements.

MATERIALS AND METHODS

In hybrid approach of Multi-objective Teacher-learning Artificial Bee colony, the software requirements for software development process are selected based on Multi-objective Teacher-Learning Artificial Bee colony optimization by means of multiple constraints and multiple objectives are cost threshold constraint, time threshold constraint and interactions constraints and minimum time consumption, maximum client satisfaction, minimum cost and maximum reliability respectively. Thus the software requirements are effectively selected by this approach.

But most of the software hazards are occurred due to incompleteness and inconsistency of requirements. Hence it is proposed that firefly technique for requirement tracing and remove the incomplete and inconsistent requirements by utilizing courteous logic-based representations of requirements. After that, complete requirements for software development is obtained and then presented a technique called semi-supervised ranking with graph-based regularization that assigned the priorities for each requirement to define which requirement is considered first for software development. Thus the software development process is carried over with more accurate requirements along with their priorities.

Requirement tracing using firefly technique

Requirement tracing is the process of trace the requirements based on life cycle of requirement and it traces the requirements from the requirement document. In the existing requirement tracing swarm techniques which are used to trace the requirement it can be enhanced by firefly algorithm for requirement tracing. For requirement tracing documents are parsed and the stop words are removed and the term frequencies and document frequencies are determined and TD-IDF weighting schema gives maximum weight to the most relevant terms within the document. The TD-IDF weighting schema is determined by the following equation:

$$TF \quad IDF_{text,doc} = TF_{text,doc} \times IDF_{text} \tag{1}$$

Where $TF_{text,doc}$ defines the count of a term occurred in a document, $IDF_{text} = log\left(\frac{number\ of\ documents}{document\ frequency}\right)$.

By using the equation (1) the documents are marked as high or low level requirement. The inverted document frequency also defines the type of the requirement. Firefly algorithm is used to allocate fixed number of agents to every requirement document with the number greater than or equal to number of requirements. The firefly algorithm depends on the attractive and intensity function (IN). Initially the populations of fireflies (number of agents) are initialized.

The intensity of fireflies is calculated as:

$$IN(i) = Minimum(False\ Error\ Rate) \tag{2}$$

The attractiveness function is calculated as:

$$\mu(y) = \mu_0 . ew^{-\delta . y^2} \tag{3}$$

where δ is the media light absorption coefficient and μ_0 is the attractiveness value of the firefly at y=0. The distance between two fireflies m and n at the position I_m and I_n are calculated as:

$$y_{mn} = \left|\left|I_m \quad I_n\right|\right| = \sqrt{\sum_{i=1}^{d}(I_{m,k} \quad I_{n,k})^2} \tag{4}$$

The movement of the firefly (m) is characterized as:

$$I_m = I_m + \mu_0 \quad \exp(\quad \delta y_{mn}^2) \quad (I_n \quad I_m) + \alpha \quad (rand \quad \tfrac{1}{2}) \tag{5}$$

Thus the movement of fireflies is determined by equation (5). By using equation (2, 3, 4, and 5) allocate fixed number of agents to every requirement document level element with the number greater than or equal to number of requirements.

Removing incompleteness

The software development process is started after the collection of unambiguous, consistent, complete, and correct requirements and these requirements are analyzed by developers, stakeholders and clients. If the requirement is satisfied, then the software development can be started. The traced requirements may contain incomplete or inconsistent requirements for software development process. Most of the software hazards are occurred due to the incomplete requirements. This can be resolved by a presented approach called logic based representation of requirements that is based on set of rules. It also improves the quality of software and it represents the requirements in logical way which is used to help in detection of incomplete and inconsistent requirements in the traced requirement. During merging, specifying and updating the rules conflicts arise it can be overcome by courteous logic programming which is an extension of logic programming, it provide solution for such conflicts. The logic program is in the form of:

$$X(0) \leftarrow X(1) \wedge X(2) \wedge \ldots \ldots .. Xi \wedge \sim X(i+1) \wedge \ldots .. \wedge \sim X(j) \tag{6}$$

In equation (6) X(y) is a literal form of an atom and ~ represents negation as failure operator X(0) is the head of the rule and followed by X (1), X(2) is the body of the rule. These rules have label before X (0) it defines the prioritization and it can be enhanced by semi-supervised ranking with graph based regularization. Thus the incomplete requirements are eliminated using the logic based representation.

Prioritization of software requirements

Prioritization is the process of taking decision about which requirement should be considered first and ordered the set of requirements for software development. In the existing, Case-Based Ranking method is processed based on three steps are pair sampling, pair elicitation and priority learning. In pair sampling, choose the set of requirements and compute the ranking of those requirements based on sampling policy. In priority elicitation gets number of sampled requirement pairs and decision maker they provide some priorities to the requirements based on the priorities, the requirements are ordered. Then in the priority learning Rank Boost algorithm is used for priority learning technique which learns the priority from the supervised labels given by priority elicitation process. These three processes are done in iterative manner until all the requirements are ranked. In this proposed work, learn the priority of the requirements from semi-supervised labels using semi-supervised ranking with graph based regularization. It considered the relative preferences of the requirements and construct an undirected graph with the nodes and edges then randomly choose the m different integers uniformly without the loss of generality. The main aim of semi-supervised ranking with graph based regularization is to create a predictor through the graph based method and evaluate the ranking performance on the unlabeled data and it defines the ranking function for the requirements.

Algorithm

Software Requirement tracing for removing incompleteness and inconsistency and prioritized software requirements

1. Calculate term frequency by counting number of terms in a document
2. Calculate inverted document frequency
   $$IDF_{text} = log\left(\frac{number\ of\ documents}{document\ frequency}\right)$$
3. Calculate the TD-IDF weighting schema
   $$TF \quad IDF_{text,doc} = TF_{text,doc} \times IDF_{text}$$
4. Trace the requirement document to requirements using firefly
5. Calculate the intensity of firefly
   $$IN(i) = Minimum(False\ Error\ Rate)$$
6. Calculate the attractiveness function
   $\mu(y) = \mu_0 . ew^{-\delta . y^2}$

```
        $I_m = I_m + \mu_0 \quad \exp( \quad \delta y_{mn}^2) \quad (I_n \quad I_m) + \alpha$
        $(rand \quad \frac{1}{2})$
7.      Construct set of rules to remove the incompleteness in
        the requirement
8.      Remove the conflicts using logic programs
        $X(0) \leftarrow X(1) \wedge X(2) \wedge \dots \dots . . Xi \wedge \sim X(i +$
        $1) \wedge \dots . . \wedge \sim X(j)$
        //prioritization
9.      Sample the set of requirements and compute the
        ranking for requirements
10.     Ordered the ranking requirements based on decision
        maker priority
                    11. $\beta(x_i, x_j) =$
        $\begin{cases} 1, if\ x_j < x_i \\ 1, if\ x_i < x_j \\ 0, no\ order\ preference\ between\ x_i\ and\ x_j \end{cases}$
        // $x_j$ and $x_i$ are the pair of requirements
12.     Priority learning based on semi-supervised ranking
        with graph-based regularization
13.     Develop a ranking framework for ranking function
        End
```

RESULTS AND DISCUSSION

In this section, the performance of existing technique is compared with proposed technique. In the existing techniques swarm technique is used for requirement and Rank Boost algorithm is used for prioritization. In the proposed technique firefly technique, is used for requirement tracing and semi-supervised ranking with graph-based regularization is used for prioritization. For the experimental purpose, 100 documents are collected from garment production management system and the performance of the proposed technique is evaluated in terms of recall, precision, F-measure.

Performance Evaluation of requirement tracing

Precision-Recall curve

Precision value is evaluated according to the requirement tracing at true positive prediction, false positive.

$$Precision = \frac{Truepositive}{(Truepositive + Falsepositive)}$$

Recall value is evaluated according to the requirement tracing at true positive prediction, false negative.

$$Recall = \frac{Truepositive}{(Truepositive + Falsenegative)}$$

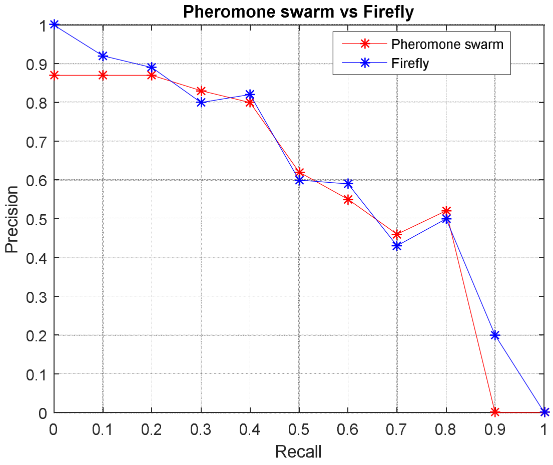

Figure.1 Precision-Recall curve between pheromone swarm and firefly

From the figure 1, it shows the proposed firefly tracing requirement technique has higher precision at 7 out of 11 recall points than the pheromone swarm technique. It is proved that the proposed technique has high precision recall curve than the existing technique.

Threshold-F measure curve

F-measure is calculated from the precision and recall value. It is calculated as:

$$f \quad measure = 2 \times \left(\frac{precision \times recall}{precision + recall}\right)$$

From the figure 2, shows the proposed firefly tracing requirement technique has high f measure value than the pheromone swarm technique at any threshold value. At 0.9 threshold value pheromone swarm shows 0.18 f measure value and at the same point firefly technique shows 0.21 measure value. It is proved that the proposed technique has high f measure than the existing technique.

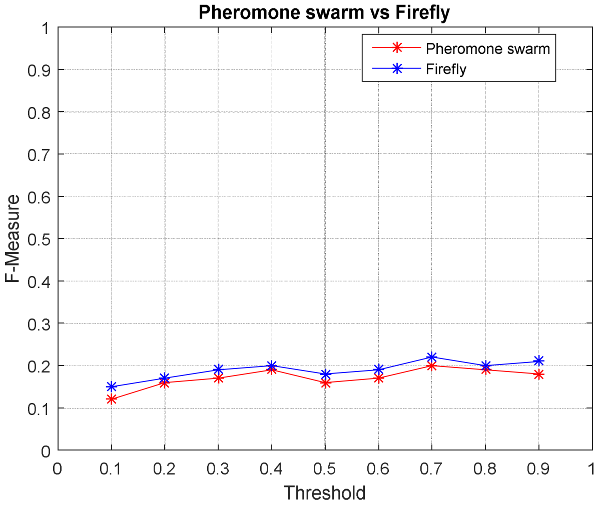

Figure.2 Threshold-F measure curve between pheromone swarm and firefly

Performance Evaluation of prioritization

Precision

Precision value is evaluated according to prioritize the requirements at true positive prediction, false positive.

$$Precision = \frac{Truepositive}{(Truepositive + Falsepositive)}$$

From the figure 3, it shows the proposed semi-supervised ranking technique has high precision value than Rank Boost technique at any number of documents. At 100 numbers of documents Rank Boost technique shows 0.89 precision and at the number of documents semi-supervised ranking technique, which shows 0.98 precision. It is proved that the proposed technique has high precision than the existing technique.

Recall

Recall value is evaluated according to prioritize the requirements at true positive prediction, false negative.

$$Recall = \frac{Truepositive}{(Truepositive + Falsenegative)}$$

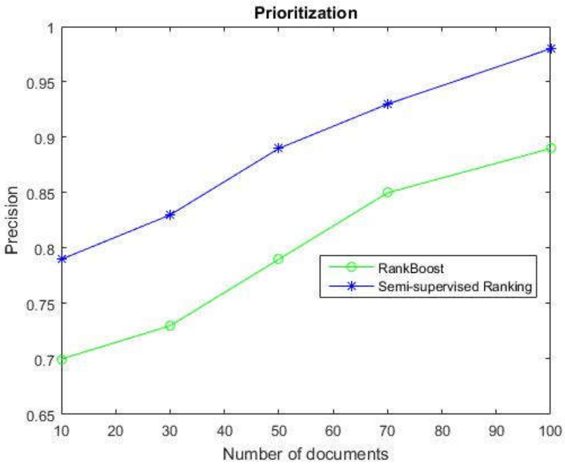

Figure.3 Precision comparison between Rank Boost and semi-supervised ranking

From the figure 4, it shows the proposed semi-supervised ranking technique has high recall value than Rank Boost technique at any number of documents. At 100 numbers of documents Rank Boost technique shows 0.88 precision and at the number of documents semi supervised ranking technique which shows 0.99 precision. It is proved that the proposed technique has high precision than the existing technique.

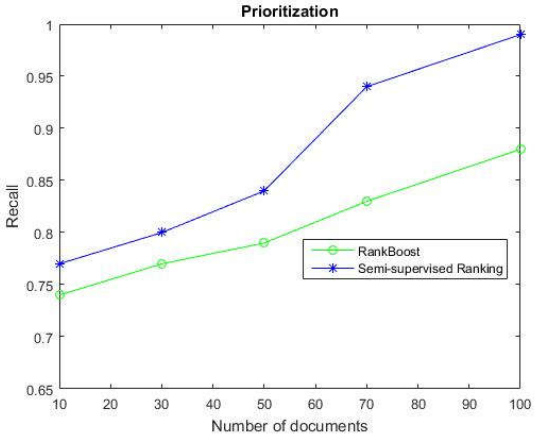

Figure.4 Recall comparison between Rank Boost and semi-supervised ranking

CONCLUSION

In software development process, requirements play a major role to improve the customer satisfaction. The requirements are collected from different customers so it may contain incomplete, ambiguous or inconsistent requirement. Moreover such requirements lead to software hazards it can be avoided by requirement tracing in which the requirement documents are traced to requirements with the help of firefly technique and remove the incomplete requirements through logic based representation of requirement. Then complete requirements of software is obtained and prioritize the requirements based on case based ranking which follows pair sampling, priority elicitation and priority learning. In this priority, learning is processed through semi-supervised ranking with graph-based regularization which is used for semi-supervised data. Finally complete requirements of software are obtained with their priorities. The experimental results are conducted to evaluate the performance of proposed technique and it is proved that the proposed techniques performed better than the existing techniques in terms of precision-recall curve, threshold-f measure curve, precision and recall.

References

Chaves-González, J. M., Pérez-Toledano, M. A., & Navasa, A. (2015). Teaching learning based optimization with Pareto tournament for the multiobjective software requirements selection. *Engineering Applications of Artificial Intelligence*, *43*, 89-101.

Elsood, M. A. A., Hefny, H. A., & Nasr, E. S. (2014, December). A goal-based technique for requirements prioritization. In *Informatics and Systems (INFOS), 2014 9th International Conference on* (pp. SW-18). IEEE.

Saraç, E., & Özel, S. A. (2013, June). Web page classification using firefly optimization. In *Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on* (pp. 1-5). IEEE.

Wang, Y., & Mao, W. (2010, November). Featurerank: A non-linear listwise approach with clustering and boosting. In *Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on* (pp. 81-84). IEEE.

Chen, S., Wang, F., Song, Y., & Zhang, C. (2011). Semi-supervised ranking aggregation. *Information Processing & Management*, *47*(3), 415-425.

Sultanov, H., Hayes, J. H., & Kong, W. K. (2011). Application of swarm techniques to requirements tracing. *Requirements Engineering*, *16*(3), 209-226.

Wang, Y., Mao, W., Zeng, D., & Bao, N. (2011, July). Boosting rank with predictable training error. In *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on* (pp. 407-409). IEEE.

Cheng, X. Q., Du, P., Guo, J., Zhu, X., & Chen, Y. (2013). Ranking on data manifold with sink points. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1), 177-191.

Perini, A., Susi, A., & Avesani, P. (2013). A machine learning approach to software requirements prioritization. *IEEE Transactions on Software Engineering*, *39*(4), 445-461.

Sharma, R., & Biswas, K. K. (2013). Resolving inconsistency and incompleteness issues in software requirements. In *Managing requirements knowledge* (pp. 245-263). Springer Berlin Heidelberg.

Yığıt, F., & Baykan, Ö. K. (2014, November). A new feature selection method for text categorization based on information gain and particle swarm optimization. In *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems* (pp. 523-529). IEEE.

Zou, Y., Ye, T., Lu, Y., Mylopoulos, J., & Zhang, L. (2015, November). Learning to Rank for Question-Oriented Software Text Retrieval (T). In*Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on* (pp. 1-11). IEEE.

Goknil, A., Kurtev, I., & Van Den Berg, K. (2014). Generation and validation of traces between requirements and architecture based on formal trace semantics. *Journal of Systems and Software*, *88*, 112-137.