Journal of Universal Computer Science, vol. 26, no. 1 (2020), 33-49
submitted: 30/12/18, accepted: 15/11/19, appeared: 28/1/20

An Intelligent Recommender System Based on Association Rule Analysis for Requirement Engineering

Mohammad Muhairat
(Al Zaytoonah University of Jordan, Amman, Jordan
drmohairat@zuj.edu.jo)

Shadi ALZu'bi
(Al Zaytoonah University of Jordan, Amman, Jordan
smalzubi@zuj.edu.jo)

Bilal Hawashin
(Al Zaytoonah University of Jordan, Amman, Jordan
b.hawashin@zuj.edu.jo)

Mohammad Elbes
(Al Zaytoonah University of Jordan, Amman, Jordan
m.elbes@zuj.edu.jo)

Mahmoud Al-Ayyoub
(Jordan University of Science and Technology, Irbid, Jordan
maalshbool@just.edu.jo)

Abstract: Requirement gathering is a vital step in software engineering. Even though many recent researches concentrated on the improvement of the requirement gathering process, many of their works lack completeness especially when the number of users is large. Data Mining techniques have been recently employed in various domains with promising results. In this work, we propose an intelligent recommender system for requirement engineering based on association rule analysis, which is a main category in Data Mining. Such recommender would contribute in enhancing the accuracy of the gathered requirements and provide more comprehensive results. Conducted experiments in this work prove that FP Growth outperformed Apriori in terms of execution and space consumption, while both methods were efficient in term of accuracy.

Key Words: Requirement Engineering, Requirements Gathering, Apriori Algorithm, FP Growth Algorithm, Association Rule Analysis, Intelligent systems, Recommender systems

Category: Topic D.2.1 - Requirements/Specifications

1 Introduction

Software development considers Requirements Engineering (RE) the most important stage in developing softwares [AlZu'bi *et al.*, 2018a, AlZu'bi *et al.*, 2018b]. It could be considered a risky project development if the RE is poorly implemented. The deriving and definition of the Requirements concentrate on requirements collection from different stakeholders. Typical resulting artifacts are textual requirement descriptions, scenario descriptions, use cases, and sketches of prototypical user interfaces. Felfernig proposed in [Felfernig *et al.*, 2013] some recommendation approaches that support activities related to requirements deriving and definition including Recommending Stakeholders, StakeNet [Lim *et al.*, 2010], Recommending Requirements, Managing Feature Requests [Mobasher and Cleland-Huang, 2011, Cleland-Huang *et al.*, 2009, Abooraig *et al.*, 2018], Consistency Management [Iyer and Richards, 2004], and Dependency Detection [Felfernig *et al.*, 2013]. Many of the previous works lack completeness especially when the number of users is large. Improving the completeness of the requirements would contribute in making the requirement gathering process successful.

Apart from this, Data Mining techniques have been recently employed in various domains [AlZu'bi *et al.*, 2019a, Alzu'bi *et al.*, 2019c, AlZu'bi *et al.*, 2019b, Mughaid *et al.*, 2019], including requirement engineering, and these methods provided promising results. One important data mining technique is the use of Recommender Systems. Recommender Systems are used to propose suggestions to users based on their interests. Recommendations are highly effective in everyday life. For example, people are mostly based on friends recommendations in reading a book or watching a movie, and in the business universe, recruitment offices are based on recommendation letters for deciding the exact persons in the hiring process. Recommender systems help in this natural and social life process to ask for referent on a given realm [Resnick and Varian, 1997, AlZu'bi *et al.*, 2019, Aloqaily *et al.*, 2019]. Recommendations generated by recommender systems can assist users sail over large information sources and/or provide propositions for items to be of use to a user (such as Amazon, Netflix, Pandora, etc.). Such systems often instruct users who do not have enough background to assess the enormous number of alternatives.

Interest in the use of recommender systems has been gown quickly in the last few years, and the software engineering community has witnessed this. This has lead to the emerge of Recommender Systems in Software Engineering (RSSE). It could be defined as a software application that provide valuable and effective suggestions for a software engineering task [Robillard and Walker, 2014]. RSSEs are now emerging to support developers in diverse activities as these systems can significantly ameliorate the accuracy of the completeness of requirement engineering [Robillard and Walker, 2014].

In this work, we propose an optimized intelligent recommender system for requirement engineering based on association rule analysis. The research objective of this work is to increase the accuracy and the completeness of the requirement engineering process by proposing such efficient recommender system for this purpose. This work has two main contributions; First, it proposes an optimized association rule based recommender system by comparing two association rule methods, namely Apriori and FP Growth. Second, this work provides a comprehensive analysis of the performance of the two methods by using three synthesized datasets of various domains and use various performance measurements. This analysis would give more insights into the promising use of data mining methods in software engineering.

The remaining sections of this paper are structured as follows. A literature for Data Mining for Requirement Engineering is reviewed in section 2. Section 3 provides more details about the proposed method. Experiments and analysis work are presented in Section 4 with the achieved results discussion. Finally, this work will be concluded in Section 6 and the road map for future work is presented.

2 Data Mining for Requirement Engineering – A Literature

Even though data mining methods have been used in various domains, few research works have used data mining methods in requirement engineering, most of which utilized the user characteristics. Felfernig et al. in [Felfernig *et al.*, 2013] provided an overview of the research dedicated to the application of recommendation technologies in RE. The authors elaborated on the use of recommender systems in various cases. They studied the knowledge-based recommendation, group-based recommendation, and social network analysis in order to suggest a future work in recommendation technologies for RE. They considered requirements elicitation and definition, quality assurance, and negotiation and release planning. For each phase, the authors provided relevant application scenarios for recommendation technologies previously identified.

Ninaus et al. showed in [Ninaus *et al.*, 2014] the support of recommender systems in the identification of related requirements in cases where the complexity of requirement assortment exceeds the user's ability [Burke, 2002, Elbes *et al.*, 2009]. They presented the following basic types of recommendation approaches:

- Collaborative Filtering: Using the well-known user-user similarity concept. This concept states that if a user likes an item, similar users would probably like the same item, and therefore, it would be suggested. [Herlocker *et al.*, 2004, Linden *et al.*, 2003, Kanan *et al.*, 2019].
- Content-based Filtering exploits item-item similarity concept. This concept states that if a user likes an item, he would probably like similar items [Pazzani and Billsus, 1997, Al-Fuqaha *et al.*, 2010]. Constraints defining the relationship between user requirements and the corresponding items is responsible for the recommendations determination.

– Group recommenders recommend items for groups of users such as recommendation of a hotel to a group of tourists who plan a common holiday trip [Felfernig *et al.*, 2011, Jameson *et al.*, 2004, Masthoff, 2011, Aqel *et al.*, 2019, Elbes *et al.*, 2019, Faqeeh *et al.*, 2014].

The authors in [Ninaus *et al.*, 2014] presented the INTELLIREQ environment that aims at making RE more proactive by the integration of various recommendation technologies. The advantages of such environment would be to increase reuse of requirements, active guidance of stakeholders, increase consistency in requirements models, and reduce time efforts needed for the construction of requirement models.

Roher and Richardson in [Roher and Richardson, 2013] used a recommender system during requirements engineering to overcome incorporating sustainability into the software engineering process. The proposed system would recommend the types of sustainability requirements that should be considered in each system based on application domain and deployment locale. The author proposed a recommender system that helps developers in requirements elicitation. The proposed system takes a hybrid approach using mainly a context-aware approach along with content-filtering algorithms. The recommender system is designed to be aware of contextual items, such as project domain and environmental factors. Based on the previous conditions, the system makes recommendations based on what the user is currently viewing, or preferences specified in search criteria. The system was evaluated using the amount of time it takes a user to discover a specific requirement archetype. Twenty users would be instructed on how to use the recommender system and asked to search for requirement archetypes that Amazon.com could use to make their e-commerce system more sustainable. Users would be observed in an interview lasting 10 minutes. The time it takes each user to discover five relevant recommendations would be recorded [Roher and Richardson, 2013].

Cleland-Huang and Mobasher studied the problem of involving huge number of stakeholders [Cleland-Huang and Mobasher, 2008]. They proposed an open, inclusive, and robust elicitation and prioritization process that utilizes data-mining and recommender technologies to facilitate the active involvement of many thousands of stakeholders. The approach claimed to be a fundamental building block towards addressing higher level requirements problems facing Ultra-Large-Scale (ULS) Systems. The proposed methodology could solve some of the problems identified in the ULS Systems report [Northrop *et al.*, 2006], such as those related to unstable requirements, emergent requirements, and variable trade-offs that occur across different instantiations of otherwise similar products.

Few common ideas have been presented by Maalej and Thurimella in [Maalej and Thurimella, 2009] for using a recommendation system in RE. The researchers studied the potentials of using recommendation systems in RE work dense action. According to the authors, the major challenge in this regard is the evaluation of the semantic similarity among collected requirements as well as the identification of problem situations and stakeholders' intents. The discovery of new requirements that are already supplied by existing frameworks and products in a large feature catalogue was a major challenge. Recommendation systems would group several requirements that are syntactically different but semantically similar. By predicting similar context, recommendation system can propose to reuse similar functional and non-functional requirements. [Hawashin *et al.*, 2019c] proposed an algorithm to extract the user interests. This extractor proved to be efficient in extracting interests and dislikes as single terms. [Hawashin *et al.*, 2019a, Mansour *et al.*, 2014a] proposed a method to extract interests of groups, as opposed to individuals, using multiple agents. This method would provide interesting statistics about the interests of the various groups. [AlZu'bi *et al.*, 2018a] proposed a method to improve user requirements by recommending correlated interests extracted by Apriori. Many solution have been proposed recently in the field of recommender systems for several applications such as in [Zarzour *et al.*, 2018, Maazouzi *et al.*, 2020, Zarzour *et al.*, 2019].

Quality assurance has been considered in [Felfernig *et al.*, 2013]. A set of requirements had to be evaluated regarding properties such as requirements' consistency, completeness of requirements by assuring that all relevant requirements should be part of the requirements model, technical and economic feasibility, fulfilling the quality standards, and reusability for future projects. Recommenders are applied to support the quality assurance for the recommendation scenarios including Recommending Stakeholders, StakeNet [Lim *et al.*, 2010], Recommending Requirements, Managing Feature Requests [Cleland-Huang *et al.*, 2009, Mobasher and Cleland-Huang, 2011], Consistency Management [Iyer and Richards, 2004], Dependency Detection [Felfernig *et al.*, 2013].

Mobasher and Cleland-Huang in [Mobasher and Cleland-Huang, 2011] used recommender systems for automating the RE processes, which enables stakeholder and designer decision support. Two objectives were targeted; The first is to identify potential stakeholders for a given project. The second is to discover user requirements or features for a system, and the third is to provide support for requirements-related decision making [Mobasher and Cleland-Huang, 2011].

A collaborative work between Systems and Requirements Engineering Center and Center for Web Intelligence introduced a new process and a related framework that utilizes recommender technologies to create an open, scalable, and inclusive requirements elicitation process capable of supporting projects. The approach was illustrated and evaluated using feature requests mined from an open source software product [Castro-Herrera *et al.*, 2008]. The researchers collected the stakeholders needs using a web-based collection tool. Clustering techniques were then employed to identify dominant and cross-cutting themes around which a set of discussion forums are created. Stakeholders were assigned to these forums based on the needs they have provided. Next, their needs were transformed into more formal requirements. The need for this type of recommender system is illustrated through examining the requirements features of open source projects. The effectiveness of the proposed recommender system was evaluated using a set of 1000 feature requests mined from SugarCRM. These feature requests were contributed by 523 different stakeholders.

There are other settings with complex inter-dependencies between requirements and many inconsistent stakeholder preferences. These settings require to adapt, combine, and extend existing recommendation approaches. One possible direction is to adapt knowledge-based recommendation functionality for group-based recommendation scenarios. Recommendation technologies will only succeed if they deliver high quality recommendations. We must design and conduct empirical studies to learn about stakeholder needs and evaluate recommendation systems. The goal is to figure out how existing recommendation approaches must be adapted for an optimal performance in RE scenarios [Felfernig *et al.*, 2013].

[Mohammadi *et al.*, 2018] proposed the use of similarities among social media users based on their interactions to predict the users who would like certain post.[Eberhard *et al.*, 2018] studied the use of labels to distinguish helpful and unhelpful reviews in Steam platform. This would help in recommending only helpful reviews to new users. [Smadi and Qawasmeh, 2018] provided an approach to extract events from Arabic tweets using supervised learning. [Mansour *et al.*, 2014b] proposed an efficient health-based recommender system for elderly people. [Hawashin *et al.*, 2019b] proposed the use of the time factor along with the user interests in the recommender systems. This would contribute in extracting the recent user interests instead of considering all interests, as some user interests could change with time.

As the field of data mining proved to be efficient in software engineering applications, it is worthwhile to use its methods for improving requirement engineering. The lack of sufficient works that study the completeness of user requirements motivated us to conduct this work.

3 Recommending Requirement Methodology

In our previous work in [AlZu'bi *et al.*, 2018a], we proposed the use of association rule discovery concepts to extract correlated user requirements. We argued that association rule discovery, which is a part of data mining that is concerned with extracting co-appearing items, can be used to find correlated requirements. We explained that these requirements may be functional or non-functional ones.

Finding such correlated requirements would help in providing recommendations to users based on their requirements. As a result, the requirements would be more comprehensive. In that work, we used one association rule method, namely Apriori, and we studied its performance based on execution time. The experimental work proved that the use of association rule discovery in requirement engineering is promising Furthermore, the experiments showed that the rule extraction time is efficient with increasing the number of users. In this work, we provide an optimized intelligent recommendations by comparing two association rule discovery methods; Apriori and FP Growth. These two methods have been used widely in the literature in various domains and proved their efficiency. Furthermore, we provide a comprehensive analysis of their performance by using various performance measurements including rule extraction time, rule lift, rule confidence, and recommendation time. In the following subsections, we provide a description of each association rule method. Next, we explain our optimized user requirement recommendation method.

3.1 Apriori

Apriori algorithm has been used widely in the literature. It is commonly used in frequent itemset mining, whereas the frequent itemsets are extracted first and the association rules are generated next. The algorithm applies Apriori principle, which states that if an itemset is not frequent, all its subsets are not frequent as well. The use of this principle has shown significant improvement in the execution time of the algorithm in comparison with the previously used comprehensive method. This algorithm was proposed by Agrawal and Srikant in 1994 [Poulain and Tarissan, 2018].

3.2 FP Growth

Frequent Pattern Growth method builds a conditional FP tree to exclude instances that do not meet the minimum support threshold. In contrast with the Apriori, this method does not generate candidate set. It scans the data twice only, in contrast with Apriori, which performs multiple scans of the data. It uses recursive processing of the compressed FP tree, which results in faster execution time in comparison with many other algorithms.

Our proposed User Requirement Recommender System Using Association Rule is given next.

3.3 requirements and the obtained rules

The following example provides a set of user requirements and the obtained rules.

Algorithm 1 User Requirement Recommender System Using Association Rule

1: **input: Set of Requirements S provided by Users**
2: **output: Set of Rules R**
3: The domain would specify the parameters of Association Rule Discovery Algorithm based on the domain needs
4: Use Association Rule Discovery Algorithm to find the set of association rules R for user requirements.
5: Return R.

System1: Resturant X
User Requirement1: The user should be able to enter the order automatically
User Requirement 2: QWERTY keyboard is used to enter requirements
System2: Resturant Y
User Requirement1: The user should be able to enter the order automatically
User Requirement 2: QWERTY keyboard is used to enter requirements
Obtained rule:
If user can enter order automatically ***then*** QWERTY keyboard is used.

Later, such rule would contribute in expanding the requirements of new users. For example, in restaurant Z, if a user requires to enter the order automatically, this rule would be used to suggest another requirement, which is to use QWERTY keyboard to enter the requirements. Obviously, these suggestions could be accepted or rejected by the user, but would contribute in enriching the list of user requirements.

4 Results and Analysis

In this section, we are going to evaluate the performance of two major association rule discovery methods; Apriori and FP Growth. In the following subsections, we are going to provide our system settings, evaluation measurements, dataset description, and experimental results.

4.1 System Settings

For our experiments, we used an Intel® Xeon® server of 3.16GHz CPU and 8GB RAM, with Microsoft Windows Server Operating System. Also, we used the implementation of WEKA 3.8.1 for Apriori and FP Growth methods. As for the preprocessing of the user requirements, we used Microsoft Visual Studio 6.

4.2 Evaluation Measurements

In order to evaluate the performance of the association rule discovery methods, we used execution time and accuracy. They are given as follows:

– Rule Extraction Time:
The time needed by the association rule discovery to extract the set of association rules from the set of requirements.
– Rule Confidence:
It is defined as the ratio of the support of rule items to the support of rule head. For a rule of the form A −¿ B, the confidence is given in the following formula.

$$Confidence(Rule) = (Support(AUB))/(Support(A)) \tag{1}$$

– Rule Lift:
It is defined as the ratio of the confidence to the support of rule head. It is given in the following formula:

$$Lift(Rule) = (Confidence(Rule))/(Support(A)) \tag{2}$$

Whereas A is the head of the rule.

– Recommendation Time:
Time needed by the recommended system to suggest new requirements based on the extracted rules.

4.3 Dataset Description

Unfortunately, there are limited number of existing resources in the literature for user requirements. We used three synthesized data of various domains. Many recent methods have been proposed to which are helpful in gathering data such as in [Ramadan *et al.*, 2019]. The description of the dataset is given in Table 1. In each dataset, each user provided one or more requirements.

Table 1: Description of the data sets

Dataset Domain	**Number of Users**	**Number of Unique Requirements**
Medical	10000	512
Library	2000	190
IT	4000	344

4.4 Experimental Results

First, as a preprocessing step, we converted all the letters to small letters. We normalized terms of various forms. As stemming could affect the requirements,

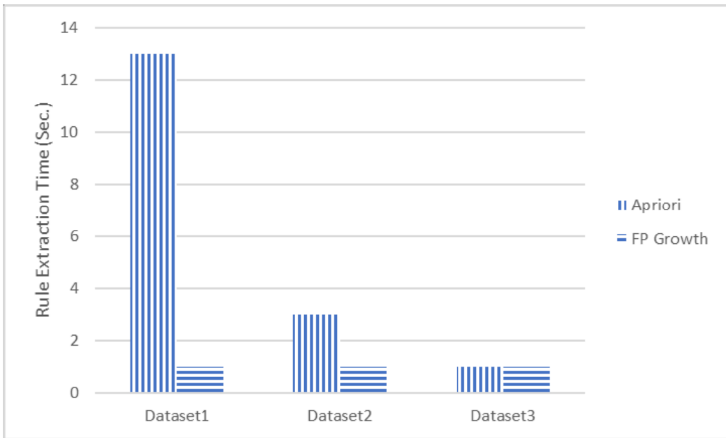

Figure 1: Comparing the rule extraction time of the two methods on the three datasets

we did not perform stemming on the terms.

Fig. 1 depicts the rule extraction time of the two methods on the three datasets. Evidently, rule extraction time varies significantly between the two methods. This difference can be noted clearly with the increase in the number of items in the dataset, as in Dataset1 that contains 10000 users. This is reasonable because FP Growth algorithm scans the data twice only, in contrast with Apriori algorithm that makes multiple scans to the data. This would contribute in increasing the rule extraction time in Apriori.

As for the space consumption, FP Growth proved to be more efficient as it does not generate candidate set, and therefore, its memory consumption would be less than that of Apriori. Regarding the recommendation time, it was negligible as the number of extracted rules is constant. Therefore, the recommendation time complexity would be O(c).

Regarding the accuracy of the method, all methods were able to extract the relevant rules that satisfy the predefined metric thresholds. The thresholds are left to the domain to specify according to its needs. Fig. 2 and Fig. 3 illustrate the average confidence and average lift of the extracted rules by the two methods on the three datasets respectively. From these figures, it is clear that both methods have similar average confidence and average lift values, and this is because of the identical extracted rules in both methods.

Fig. 4 and Fig. 5 depict the effect of changing minimum support threshold and minimum confidence on the rule extraction time of the two methods on the three datasets respectively. It is interesting to see that Apriori rule extraction time is

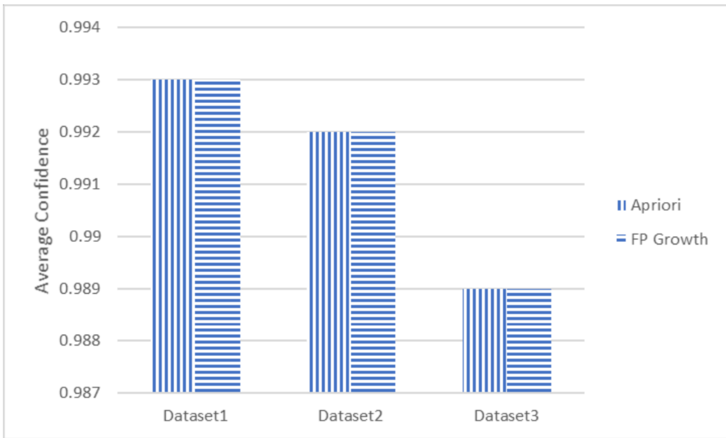

Figure 2: Comparing the average confidence of the two methods on the three datasets

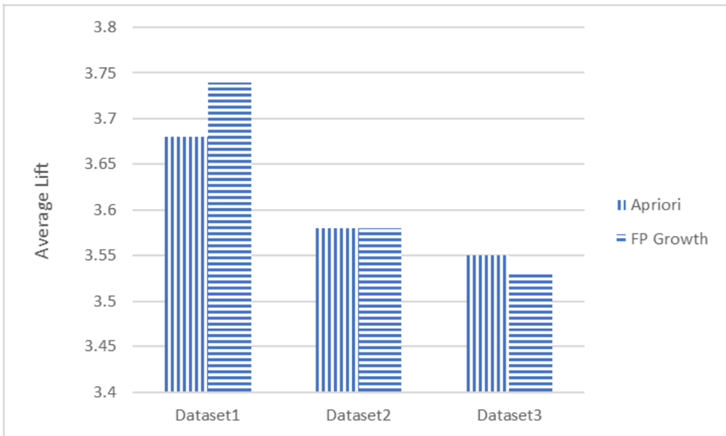

Figure 3: Comparing the average Lift of the two methods on the three datasets

significantly affected by the minimum support threshold, but it is not affected by the minimum confidence level. This is due to the fact that the minimum support plays an important rule in generating candidates in the first phase of Apriori, which is a time consuming phase, while the minimum confidence plays an important role in the second phase of Apriori, which is the rule generation, and this phase is not that time consuming. As for FP Growth, its performance was stable and efficient with changing support and confidence values.

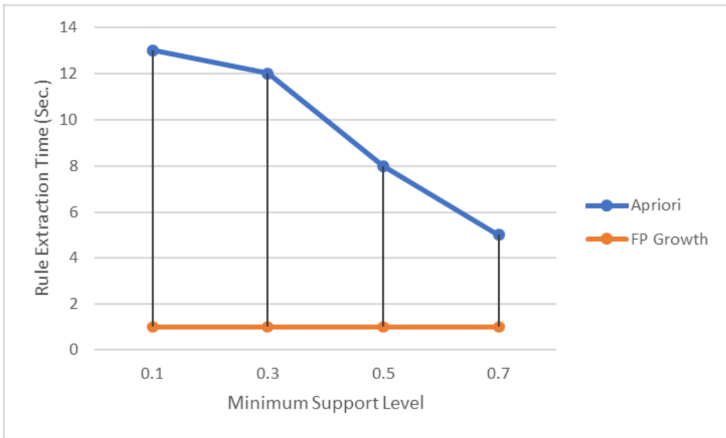

Figure 4: Rule extraction time of the two methods on Dataset 1 with changing minimum support level

5 Discussion

From the previous experiments, one of the key observations is that FP Growth algorithm proved to be very fast in extracting rules, in contrast with Apriori algorithm. This can be seen clearly when the dataset is large, as Apriori needs multiple scans of the data set, while FP Growth performs two scans only.

Regarding the the proposed system accuracy, it is observed that both Apriori and FP Growth showed similar performance and were comparable in terms of confidence and lift.

One of the drawbacks of the proposed recommender system is the need to specify the threshold for the minimum support and the minimum accuracy levels. It was observed that these thresholds could highly affect the results and it is necessary to the optimal values for these thresholds. Moreover, it was noted that some association methods are more affected by the change in these thresholds than other methods. This part can be extended and analyzed more thoroughly in the future.

Furthermore, more association rule discovery methods could be compared to further improve the performance.

Finally, more data mining techniques could be used in requirement engineering as many promising methods were not utilized yet in requirement engineering.

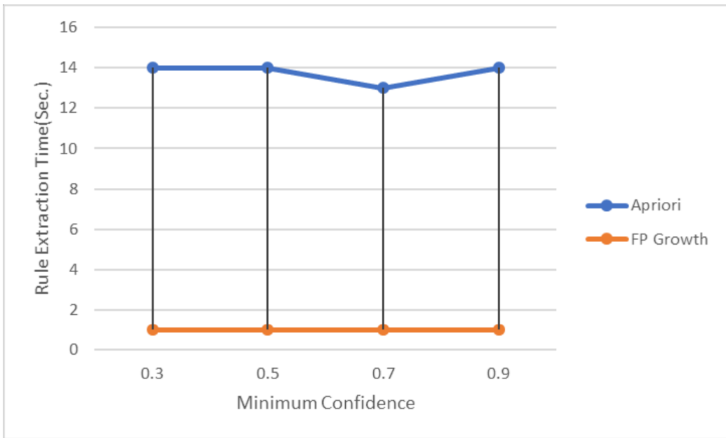

Figure 5: Rule extraction time of the two methods on Data set 1 with changing minimum confidence

6 Conclusions

In this work, an intelligent recommender system for requirement engineering based on association rule analysis has been proposed to enhance the accuracy and the completeness of the gathered requirements and provide more comprehensive results. Conducted experiments in this work showed that FP Growth outperformed Apriori in terms of rule extraction time and space consumption, while both methods were superior in term of accuracy.

Future work can be conducted in various directions. On one hand, more association rule algorithms could be used to further improve the performance. On the other hand, it is necessary to find the optimal threshold values for the minimum support and the minimum accuracy levels.

Finally, it would be interesting to use more data mining techniques in requirement engineering as many promising methods were not utilized yet in this field.

References

[Abooraig *et al.*, 2018] Raddad Abooraig, Shadi Al-Zu'bi, Tarek Kanan, Bilal Hawashin, Mahmoud Al Ayoub, and Ismail Hmeidi. Automatic categorization of arabic articles based on their political orientation. *Digital Investigation*, 25:24–41, 2018.

[Al-Fuqaha *et al.*, 2010] A Al-Fuqaha, D Kountanis, S Cooke, M Elbes, and J Zhang. A genetic approach for trajectory planning in non-autonomous mobile ad-hoc networks with qos requirements. In *2010 IEEE Globecom Workshops*, pages 1097–1102. IEEE, 2010.

[Aloqaily *et al.*, 2019] Moayad Aloqaily, Ismaeel Al Ridhawi, Haythem Bany Salameh, and Yaser Jararweh. Data and service management in densely crowded environments: Challenges, opportunities, and recent developments. *IEEE Communications Magazine*, 57(4):81–87, 2019.

[AlZu'bi *et al.*, 2018a] S. AlZu'bi, B. Hawashin, M. Elbes, and M. Al-Ayyoub. A novel recommender system based on apriori algorithm for requirements engineering. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 323–327, Oct 2018.

[AlZu'bi *et al.*, 2018b] Shadi AlZu'bi, Sokyna Al-Qatawneh, and Mohammad Alsmirat. Transferable hmm trained matrices for accelerating statistical segmentation time. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 172–176. IEEE, 2018.

[AlZu'bi *et al.*, 2019a] Shadi AlZu'bi, Abdalraheem Alsmadiv, Sokyna AlQatawneh, Mahmoud Al-Ayyoub, Bilal Hawashin, and Yaser Jararweh. A brief analysis of amazon online reviews. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 555–560. IEEE, 2019.

[AlZu'bi *et al.*, 2019b] Shadi AlZu'bi, Darah Aqel, Alaa Mughaid, and Yaser Jararweh. A multi-levels geo-location based crawling method for social media platforms. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 494–498. IEEE, 2019.

[Alzu'bi *et al.*, 2019c] Shadi Alzu'bi, Omar Badarneh, Bilal Hawashin, Mahmoud Al-Ayyoub, Nouh Alhindawi, and Yaser Jararweh. Multi-label emotion classification for arabic tweets. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 499–504. IEEE, 2019.

[AlZu'bi *et al.*, 2019] Shadi AlZu'bi, Bilal Hawashin, Muhannad Mujahed, Yaser Jararweh, and Brij B Gupta. An efficient employment of internet of multimedia things in smart and future agriculture. *Multimedia Tools and Applications*, pages 1–25, 2019.

[Aqel *et al.*, 2019] Darah Aqel, Shadi AlZu'bi, and Siham Hamadah. Comparative study for recent technologies in arabic language parsing. In *2019 Sixth International Conference on Software Defined Systems (SDS)*, pages 209–212. IEEE, 2019.

[Burke, 2002] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[Castro-Herrera *et al.*, 2008] Carlos Castro-Herrera, Chuan Duan, Jane Cleland-Huang, and Bamshad Mobasher. Using data mining and recommender systems to facilitate large-scale, open, and inclusive requirements elicitation processes. In *International Requirements Engineering, 2008. RE'08. 16th IEEE*, pages 165–168. IEEE, 2008.

[Cleland-Huang and Mobasher, 2008] Jane Cleland-Huang and Bamshad Mobasher. Using data mining and recommender systems to scale up the requirements process. In *Proceedings of the 2nd international workshop on Ultra-large-scale software-intensive systems*, pages 3–6. ACM, 2008.

[Cleland-Huang *et al.*, 2009] Jane Cleland-Huang, Horatiu Dumitru, Chuan Duan, and Carlos Castro-Herrera. Automated support for managing feature requests in open forums. *Communications of the ACM*, 52(10):68–74, 2009.

[Eberhard *et al.*, 2018] Lukas Eberhard, Patrick Kasper, Philipp Koncar, and Christian Gütl. Investigating helpfulness of video game reviews on the steam platform. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 43–50. IEEE, 2018.

[Elbes *et al.*, 2009] Mohammed Elbes, Ala Al-Fuqaha, Mohsen Guizani, Ammar Rayes, and Jun S Oh. A new hierarchical and adaptive protocol for minimum-delay v2v communication. In *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pages 1–6. IEEE, 2009.

[Elbes *et al.*, 2019] Mohammed Elbes, Shadi Alzubi, Tarek Kanan, Ala Al-Fuqaha, and Bilal Hawashin. A survey on particle swarm optimization with emphasis on engineering and network applications. *Evolutionary Intelligence*, pages 1–17, 2019.

[Faqeeh *et al.*, 2014] Mosab Faqeeh, Nawaf Abdulla, Mahmoud Al-Ayyoub, Yaser Jararweh, and Muhannad Quwaider. Cross-lingual short-text document classification for facebook comments. In *2014 International Conference on Future Internet of Things and Cloud*, pages 573–578. IEEE, 2014.

[Felfernig *et al.*, 2011] Alexander Felfernig, Christoph Zehentner, Gerald Ninaus, Harald Grabner, Walid Maalej, Dennis Pagano, Leopold Weninger, and Florian Reinfrank. Group decision support for requirements negotiation. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 105–116. Springer, 2011.

[Felfernig *et al.*, 2013] Alexander Felfernig, Gerald Ninaus, Harald Grabner, Florian Reinfrank, Leopold Weninger, Denis Pagano, and Walid Maalej. An overview of recommender systems in requirements engineering. In *Managing requirements knowledge*, pages 315–332. Springer, 2013.

[Hawashin *et al.*, 2019a] Bilal Hawashin, Shadi Alzubi, Tarek Kanan, and Ayman Mansour. An efficient semantic recommender method forarabic text. *The Electronic Library*, 37(2):263–280, 2019.

[Hawashin *et al.*, 2019b] Bilal Hawashin, Darah Aqel, Shadi AlZu'bi, and Yaser Jararweh. Novel weighted interest similarity measurement for recommender systems using rating timestamp. In *2019 Sixth International Conference on Software Defined Systems (SDS)*, pages 166–170. IEEE, 2019.

[Hawashin *et al.*, 2019c] Bilal Hawashin, Ayman Mansour, Jafar Abukhait, Fayez Khazalah, Shadi AlZu'bi, Tarek Kanan, Mohammad Obaidat, and Mohammed Elbes. Efficient texture classification using independent component analysis. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 544–547. IEEE, 2019.

[Herlocker *et al.*, 2004] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[Iyer and Richards, 2004] Jyothi Iyer and Debbie Richards. Evaluation framework for tools that manage requirements inconsistency. In *Proceedings of the 9th Australian Workshop on Requirements Engineering (AWRE'04)*, 2004.

[Jameson *et al.*, 2004] Anthony Jameson, Stephan Baldes, and Thomas Kleinbauer. Two methods for enhancing mutual awareness in a group recommender system. In *Proceedings of the working conference on Advanced visual interfaces*, pages 447–449. ACM, 2004.

[Kanan *et al.*, 2019] Tarek Kanan, Odai Sadaqa, Amal Aldajeh, Hanadi Alshwabka, Shadi AlZu'bi, Mohammed Elbes, Bilal Hawashin, Mohammad A Alia, et al. A review of natural language processing and machine learning tools used to analyze arabic social media. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 622–628. IEEE, 2019.

[Lim *et al.*, 2010] Soo Ling Lim, Daniele Quercia, and Anthony Finkelstein. Stakenet: using social networks to analyse the stakeholders of large-scale software projects. In *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pages 295–304. ACM, 2010.

[Linden *et al.*, 2003] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

[Maalej and Thurimella, 2009] Walid Maalej and Anil Kumar Thurimella. Towards a research agenda for recommendation systems in requirements engineering. In *Managing Requirements Knowledge (MARK), 2009 Second International Workshop on*, pages 32–39. IEEE, 2009.

[Maazouzi *et al.*, 2020] Faiz Maazouzi, Hafed Zarzour, and Yaser Jararweh. An effective recommender system based on clustering technique for ted talks. *International Journal of Information Technology and Web Engineering (IJITWE)*, 15(1):35–51, 2020.

[Mansour *et al.*, 2014a] Ayman M Mansour, Mohammad A Obaidat, and Bilal Hawashin. Elderly people health monitoring system using fuzzy rule based approach. *International Journal of Advanced Computer Research*, 4(4):904, 2014.

[Mansour *et al.*, 2014b] Ayman M Mansour, Mohammad A Obaidat, and Bilal Hawashin. Elderly people health monitoring system using fuzzy rule based approach. *International Journal of Advanced Computer Research*, 4(4):904, 2014.

[Masthoff, 2011] Judith Masthoff. Group recommender systems: Combining individual models. In *Recommender systems handbook*, pages 677–702. Springer, 2011.

[Mobasher and Cleland-Huang, 2011] Bamshad Mobasher and Jane Cleland-Huang. Recommender systems in requirements engineering. *AI magazine*, 32(3):81–89, 2011.

[Mohammadi *et al.*, 2018] Samin Mohammadi, Reza Farahbakhsh, and Noel Crespi. Who will like the post? a case study of predicting likers on flickr. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 35–42. IEEE, 2018.

[Mughaid *et al.*, 2019] Ala Mughaid, Ibrahim Obeidat, Bilal Hawashin, Shadi AlZu'bi, and Darah Aqel. A smart geo-location job recommender system based on social media posts. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 505–510. IEEE, 2019.

[Ninaus *et al.*, 2014] Gerald Ninaus, Alexander Felfernig, Martin Stettinger, Stefan Reiterer, Gerhard Leitner, Leopold Weninger, and Walter Schanil. Intellireq: Intelligent techniques for software requirements engineering. In *ECAI*, pages 1161–1166, 2014.

[Northrop *et al.*, 2006] Linda Northrop, Peter Feiler, Richard P Gabriel, John Goodenough, Rick Linger, Tom Longstaff, Rick Kazman, Mark Klein, Douglas Schmidt, Kevin Sullivan, et al. Ultra-large-scale systems: The software challenge of the future. Technical report, Carnrgie-Mellon University Pittsburgh PA software engineering institute, 2006.

[Pazzani and Billsus, 1997] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331, 1997.

[Poulain and Tarissan, 2018] Rémy Poulain and Fabien Tarissan. Quantifying the diversity in users activity: An example study on online music platforms. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 3–10. IEEE, 2018.

[Ramadan *et al.*, 2019] Rana Ramadan, Sokyna Alqatawneh, Fadwa Ahalaiqa, Ikhals Abdel-Qader, Ali Aldahoud, and Shadi AlZoubi. The utilization of whatsapp to determine the obsessive-compulsive disorder (ocd): A preliminary study. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 561–564. IEEE, 2019.

[Resnick and Varian, 1997] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[Robillard and Walker, 2014] Martin P Robillard and Robert J Walker. An introduction to recommendation systems in software engineering. In *Recommendation Systems in Software Engineering*, pages 1–11. Springer, 2014.

[Roher and Richardson, 2013] Kristin Roher and Debra Richardson. A proposed recommender system for eliciting software sustainability requirements. In *User Evaluations for Software Engineering Researchers (USER), 2013 2nd International Workshop on*, pages 16–19. IEEE, 2013.

[Smadi and Qawasmeh, 2018] Mohammad Smadi and Omar Qawasmeh. A supervised machine learning approach for events extraction out of arabic tweets. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 114–119. IEEE, 2018.

[Zarzour *et al.*, 2018] Hafed Zarzour, Ziad Al-Sharif, Mahmoud Al-Ayyoub, and Yaser Jararweh. A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. In *2018 9th International Conference on Information and Communication Systems (ICICS)*, pages 102–106. IEEE, 2018.

[Zarzour *et al.*, 2019] Hafed Zarzour, Ziad A Al-Sharif, and Yaser Jararweh. Recdnning: a recommender system using deep neural network with user and item embeddings. In *2019 10th International Conference on Information and Communication Systems (ICICS)*, pages 99–103. IEEE, 2019.