IPTEK The Journal of Technology and Science, 32(3), 2021 (e/pISSN:2088-2033, 0853-4098)

DOI: 10.12962/j20882033.v32i3.13003 | Received 11 September, 2021; Accepted 8 Dec, 2021

ORIGINAL RESEARCH

A SEMANTIC COMPARISON OF FEATURE REQUIREMENTS EXTRACTION METHODS

Patricia Gertrudis Manek*[1] | Abdullah Faqih Septiyanto[2] | Adi Setyo Nugroho[2]

[1]Dept. of Information Technology, Universitas Timor, Kefamenamu, Indonesia

[2]Informatics Master Programme, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Correspondence

*Patricia Gertrudis Manek, Dept of Information Technology, Universitas Timor, Kefamenamu, Indonesia. Email: gertrudismanek@unimor.ac.id

Present Address

Gedung Fakultas Pertanian, Jl. KM. 9,
Kefamenamu, Timor Tenagh Utarra 85613,
Indonesia

Abstract

Requirement engineering is an essential part of software development. The initial process in software development is to determine the needs of the stakeholders. To convert stakeholder needs into features of the system to be developed takes a long time, so it is a challenge for researchers to be able to extract features automatically based on the description of the needs of stakeholders. Previous research has also implemented feature extraction using user reviews on applications that public users have used. The feature extraction results will be used for feature development in future updated versions. The extraction process can use several proven methods to provide results that match the needs of the stakeholders in the system. This study compared the automatic feature extraction method using Natural Language Processing (NLP) with Hierarchical Pattern Recognition (HPR) on the dataset requirements and user reviews. Performance evaluation was conducted to test feature extraction results using Accuracy, precision, recall, and F-measure. The study results show that each method has advantages when implemented on both datasets. The NLP method excels in classifying the NL Requirement dataset. The HPR method has its advantages in extracting user review data.

KEYWORDS:

Feature Extraction, Natural Language, Pattern Recognition, Requirement Engineering

1 | INTRODUCTION

Requirements engineering is an activity in conducting needs elicitation, analysis, evaluation, and documentation of needs. Requirements engineering is the main foundation in system development, where failure at this phase can cause a project to fail. According to research, most project failures are in the needs engineering phase[1]. Requirements engineering is an essential part of software development. Some scholars suggest that developing software is not merely about a technological issues but also involve collaboration and communication skills between different stakeholders[2]. So engineers must have communication skills with balanced engineering knowledge[3]. Based on these problems, to get the needs of stakeholders precisely and efficiently, automatic feature extraction is applied as in previous studies.

In requirements engineering, documented requirements will continue to become a system's feature. So a feature extraction process is carried out to dig up information on needs easily and more efficiently. Extract features semi-automatically. It is done using Natural Language Processing (NLP) methods because the system requirement documents are written in natural language. Several previous studies in feature extraction were carried out on two objects[4]. The first is feature extraction on the Software Requirements Specification (SRS) document. The SRS document is an artefact that contains details regarding functional and non-functional requirements. The second is feature extraction on user opinions or software reviews. There is a lot of research in this field because user opinions can be considered by developers when doing further development. And also, because SRS documents are confidential and not everyone has access to them.

In previous research, Haris conducted research on automatic requirement extraction on the Software Specification Document (SRS) Document as the basis of the software product line[5]. This study focuses on using NLP to extract sentences based on boilerplate templates with the Part-of-Speech (POS) Tagging method, which will later become the basis of the feature. Putri and Siahaan[6] conducted research to extract data from software opinion documents from 3 app store applications. They modified the collocation method by analyzing the dependence between words to get features that are rarely mentioned. Then another study was conducted by Bakar et al.[7] to yank software capabilities from software reviews, which are publicly accessible, to prevent need redundancies. It uses feature extraction natural language (FENL) to yank terms from review sentences to generate software features.

Based on the three studies, each study uses a different type of dataset, namely using the SRS document or user review/user opinion. To find out what method is the most optimal for performing feature extraction for each type of dataset, the researcher makes comparisons for each method in previous research using the same type of dataset. Research using the SRS dataset is grouped, and research using user review datasets is also grouped. Each group was compared in performing feature extraction so that the highest precision, recall, and f-measure values of each method with the same dataset could be seen. So that we find the best method for extracting the features of the two groups.

2 | PREVIOUS RESEARCHES

Several studies have carried out a semi-automatic and automatic feature extraction using SRS documents in related research. Research using SRS documents has differences. Some use a list of functional requirements based on SRS documents. Some use full SRS documents. Research on feature extraction using the SRS document was successfully carried out with different accuracy values. Haris et al.[5] process SRS documents using the Natural Language Processing (NLP) method with boilerplate requirements as a statement of the sentence requirements. Automatic identification and extraction of statement sentences using Part-of-Speech (POS) sequences. The research has used public SRS documents. Siahaan et al.[8] introduce a tool to extract semi-automatic functional and non-functional features in SRS documents into metadata in RDF files. The method used is POS with TF * IDF. POS is used for tagging, and TF*IDF is used for extraction requirements. Haque et al.[9] combines feature extraction with machine learning to classify Non Functional Requirements (NFR). Testing was carried out using seven machine learning algorithms with four feature selection approaches to find the best pair for performing feature extraction and classification. The Stochastic Gradient Descent Support Vector Machine (SFD SVM) got the best results of all the experiments conducted.

Feature extraction using SRS documents has data limitations because not everyone can quickly access or get SRS documents. Several studies conducted feature extraction using user reviews. There are many responses from users when using software or applications in user reviews. So from these data, feature extraction can be carried out to be used as a reference for future feature improvements. Putri and Siahaan[10] made improvements in extracting less frequently mentioned software features based on end-user review. To enhanced the result produced by the collocation finding methods, the etymological rules were added. Furthermore, to avoid extracting minor relevant features, the feature trimming was also added. The results of this study are better than the collocation find the method, with more feature extraction results.

Htay and Lynn[11] extract features from user review data by obtaining opinion word/phrase patterns through adjectives, adverbs, verbs, and nouns. POS-tagging is used to create general language patterns, parse sentences, and identify product features

TABLE 1 The experimental dataset: requirements.

Project ID	Number of Rows
1	20 Rows
2	21 Rows
3	43 Rows
4	30 Rows
5	30 Rows
6	30 Rows
7	14 Rows
8	9 Rows
9	54 Rows
10	42 Rows
11	3 Rows
12	8 Rows
13	2 Rows
14	3 Rows
Total	309 Rows

and positive or negative opinion words. Then all opinions are summarized and grouped according to their orientation. Bakar et al.[12] proposed a semi-automatic software feature extraction method based on online reviews to assist in reusing natural language requirements. The NLP method is applied using the information retrieval technique. The semi-automatic extraction result compared to manual extraction shows that the estimated time needed to extract features is faster using the semi-automatic.

Raharjana et al.[13] provide an update from previous research on feature extraction and sentiment analysis with similarity measures from user reviews for reuse requirements. The extracted features are clustered based on polarity, subjectivity, and similarity values. This clusters are used for evaluating the association between the three values with the results of software feature extraction. The cluster that has positive sentiment should have better outcome than the cluster that has negative sentiment with a high similarity value. Lastly, Hasrina et al.[14] extracts software features of numerous kinds, such as legacy functionalities, software reviews, or software description which is processed using natural language processing (NLP) and information retrieval methods.

3 | MATERIAL AND METHOD

In this research, the methods used are Natural Language Processing (NLP) and Pattern Recognition Hierarchical (HPR). Researchers use two datasets for comparison testing, namely the SRS document containing NLP requirements and user reviews. NLP and HPR will be used to extract features from both datasets. The two datasets are used to divide the testing group of several methods in previous research.

3.1 | Natural Language Software Requirement

Here, we are using one of the previous research datasets that can be accessed via GitHub (https://github.com/tobhey/NoRBERT) called NoRBERT dataset. This dataset contains the functional requirements data from the PROMISE dataset. There are 309 rows of data of necessity from 14 software projects. The requirements dataset consists of several columns, namely ProjectID, RequirementText, Function, Data, and Behavior. ProjectID shows the numeric id of the project, RequirementText shows the requirement statement, and Function shows the requirement statement belongs to the required function class. The data indicates that the requirement statement belongs to the requirements data class, and Behavior shows the requirement statement, including the requirement members that can benefit the user. Each requirement text is grouped into each project id shown in Table 1 .

From Table 1, the requirement dataset is preprocessed so that the dataset can be used in the feature extraction method. Preprocessing is done to get the total verb, noun, word, and sentence as statistics. Because the data distribution is needed to map how many features are in the dataset. In general, features are obtained from verbs. The results of the preprocessing requirements datasets are shown in Figure 1 .

3.2 | User Reviews

The second dataset we use is the google play apps user review dataset. The dataset is obtained from kaggle (https://www.kaggle.com/yassershrief/goggle-play-data), consisting of user reviews on the application. The number of review

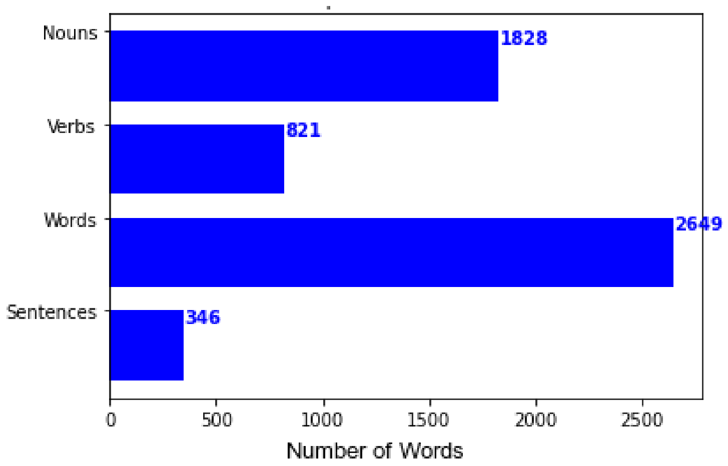


FIGURE 1 The result of pre-processing the requirement dataset.

TABLE 2 The experimental dataset: user reviews.

Project ID	Number of Rows
1	113 Rows
2	39 Rows
3	39 Rows
4	62 Rows
5	47 Rows
Total	300 Rows

records on this dataset is probably 64.295. The user reviews dataset consists of several columns: App, Translated_review, and Sentiment. The app contains the name of the app that has been reviewed. Translated_review includes user reviews that have been translated into English. And sentiment includes review categorization, positive and negative. From some of these components, we only take the review sentence to be used as material for research. The total record of this dataset is reduced because this dataset doesn't contain any ground truth. Only review from 5 applications that have a total of 300 records. Finally, each translated review is grouped into each project id shown in Table 2. This review dataset is also preprocessed like the NL requirements dataset before. We find a statistic about how many verbs, nouns, words, and sentences are inside the dataset. The result of preprocessing reviews dataset are shown in Figure 2 .

In our research, the final result obtained is a feature extraction from two datasets, namely NLP Requirements and User Review. The review results will then be re-examined to determine whether it has succeeded in extracting features according to the dataset. Some of the assessment metrics that we use include Precision (PR), Recall (RC), Accuracy (AC), and F1-Score (F1) using a confusion matrix. But before we calculate the metric, we first need to find parameters TP, TF, FP, and FN. Figure. Three shows describe all of those parameters.

After finding all those four parameters, all metrics can be found by calculating them with a formula. Accuracy is the most critical metric because we need almost all combinations of parameters to calculate it. Equation 1 describes how to calculate Accuracy.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

We defined precision as the correctness percentage of the confirmed result to the total predicted positive. Typically, high precision relates to the low false-positive rates. The precision calculation can be done using Eq. 2.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

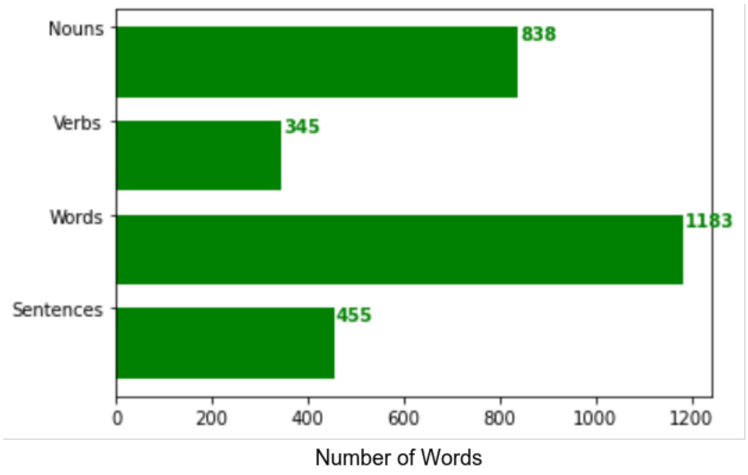
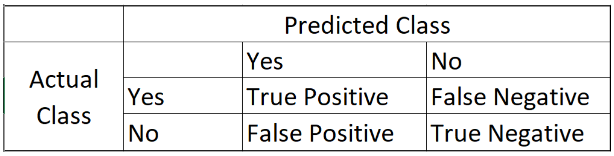


FIGURE 2 The result of pre-processing the user review dataset.



	Predicted Class		
Actual Class		Yes	No
	Yes	True Positive	False Negative
	No	False Positive	True Negative

FIGURE 3 The confusion matrix.

The recall is a correctness ratio of a positive result to all class yes. The calculation of recall seems the same as before but uses False Negative. Equation 3 is used to calculate recall.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The last metric of calculation is the F1 score. F1 score does not require value from parameter before, but the result of precision and recall. The score represents a weighted mean between precision and recall. F1 score is not easy to understand but usually more helpful than accuracy metrics. Because it is had advantages when FP and FN values are very different. F1 score calculation is done by Eq. 4.

$$F1Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{4}$$

4 | RESULTS AND DISCUSSION

The performance of the method was evaluated from the natural language processing method. We implement these methods using the Python programming language. We used the method used in previous studies, mostly carried out, namely Part of Speech Tagging (POS Tagging) and Subject Verb Object Analysis with Pattern Recognition. Preprocessing is done to get a dataset that can be processed using NLP and HPR. Some of the techniques used for preprocessing are stopword removal, punctuation, removing special characters, and tokenizing. After the preprocessing stage, the data is ready to be processed for feature extraction using Part of Speech Tagging (POS Tagging) based on NLP and HPR. The results of feature extraction from the two methods were compared. Both methods can perform feature extraction on the requirements dataset and user reviews dataset used for testing in this research. The total features extracted by the NLP and HPR methods are shown in Table 3.

TABLE 3 Feature Extraction Comparison.

ID	Requirements Dataset			User Reviews Dataset		
	#Sentence	#NLP	#HPR	#Sentence	#NLP	#HPR
1	20	19	19	113	62	57
2	22	21	22	39	13	9
3	44	44	44	39	26	22
4	45	45	26	62	34	40
5	42	41	37	47	91	45
6	33	33	31			
7	15	14	15			
8	42	42	38			
9	17	17	17			
10	45	45	43			
11	3	3	3			
12	8	8	8			
13	6	6	3			
14	4	4	3			

TABLE 4 Performance evaluation comparison of NLP and HPR.

ID	NLP				HPR			
	AC	PR	RC	F1	AC	PR	RC	F1
	Requirements Dataset							
1	0,95	1	0,95	0,97	0,95	1	0,95	0,97
2	0,95	1	0,95	0,97	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	0,57	1	0,57	0,73
5	0,97	1	0,97	0,98	0,88	1	0,88	0,93
6	1	1	1	1	0,93	1	0,93	0,96
7	0,93	1	0,93	0,96	1	1	1	1
8	1	1	1	1	0,9	1	0,9	0,95
9	1	1	1	1	1	1	1	1
10	1	1	1	1	0,95	1	0,95	0,97
11	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1
13	1	1	1	1	0,5	1	0,5	0,66
14	1	1	1	1	0,75	1	0,75	0,85
	Requirements Dataset							
1	0,69	0,17	0,84	0,29	0,56	0,17	0,83	0,28
2	0,76	0,46	0,60	0,52	0,89	0,78	0,78	0,78
3	0,64	0,38	0,58	0,46	0,68	0,52	0,84	0,64
4	0,67	0,32	0,47	0,38	0,5	0,35	0,73	0,47
5	0,53	0,23	0,72	0,35	0,38	0,36	1	0,52

From Table 3 , it can be seen that each dataset processed by this method has different feature extraction results but not far enough. In the first and second datasets, the features produced by the POS Tagging method are more than those of the HPR. This is because the POS Tag method has a lot of rules that perform an in-depth analysis of each sentence in the dataset.

The extraction results with the number of sentences in the first and second datasets are slightly different. This can be seen in the results of both methods. In the first dataset, the results obtained are only somewhat different from the number of original sentences. Compared to the second dataset, the results are pretty far apart. The factor that causes these differences is the standard of each dataset. In the first dataset, the dataset has been neatly arranged according to the standard for writing software requirements, such as the standard Boiler Plate. At the same time, the second dataset is quite broad in scope because it is the opinion of the users. There are many ambiguous sentence writing and non-standard abbreviations, which make the detection of needs less than optimal.

After extracting the feature from the dataset, we calculated metrics using Equations 1 – 4 from the previous section to evaluate performance from both methods. The result of the evaluation can be seen in Table 4 .

In this research, we grouped the two datasets by their applications. In the first dataset, there are 14 applications, while in the second dataset, we cut it down to 5 applications, each of which has 300 records. Likewise, we also calculate Accuracy, precision, recall, and F1 score for each application per dataset for metric calculations. In Table 4, it can be seen that the first dataset has an oddity, which is in the precision value. The precision value in the first dataset using both methods result in 1. This is because the dataset is a feature of the system, which has been arranged in a clean sentences format. That's why the number of FP directly correlates with precision is 0. Meanwhile, in the second dataset, user comments tend to be random and sometimes contain criticism, which causes not all sentences to be classified as features

When viewed from the comparison of method results, the POS Tag method has the advantage of performing feature extraction on the first dataset. This can be seen from the number of 1 values in Table 7. This is due to the nature of the dataset and the method's ability to form rules so that it can extract features well. In comparison, the HPR method has the advantage of feature extraction on the user review dataset. The advantage of this method is that this method is more straightforward, does not have many rules, and focuses on the main verb.

5 | CONCLUSION

Requirement engineering has been one of the critical research in the last decade. Because understanding what users need is a critical thing to measure project success. Several researchers in requirement engineering expertise have conducted some methods to perform feature extraction in natural language. Feature extraction in natural language is beneficial to find the basic knowledge from the requirement sentences.

This research conducted a comparison study in 2 methods to extract features from 2 natural language datasets. Our research results show that both methods can perform feature extraction very well. NLP method can extract features better in the requirement dataset. Meanwhile, the HPR method has the best result in extracting features in User Review Dataset. Both methods have positive and negative when implemented in each dataset.

To further understand the causes of the results of this study, further studies are needed to improve extraction results or improve existing NLP and HPR methods. NLP can be enhanced using the new POS TAGGING merging rules to detect more flexible features in various sentence structures. HPR can be increased by adding a new extraction process in it. Ambiguous sentence handling can also be applied to extract sentences with multiple structures.

CREDIT

Patricia Getrudis Manek: Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing, and Supervison. **Abdullah Faqih Setiyanto:** Data Curation, Writing - Original Draft, Validation, and Investigation. **Adi Setyo Nugroho:** Data Curation, Writing - Original Draft, Validation, and Investigation.

References

1. Ghozali RP, Saputra H, Nuriawan MA, Suharjito, Utama DN, Nugroho A. Systematic literature review on decision-making of requirement engineering from agile software development. In: Procedia Computer Science, vol. 157 Elsevier B.V.; 2019. p. 274–281.
2. Grunbacher P, Seyff N. Requirements Negotiation. Engineering and Managing Software Requirements 2005;p. 143–162.
3. Alsanoosy T, Spichkova M, Harland J. Exploratory Analysis of Cultural Influences on Requirements Engineering Activities Based on Stakeholders' Profile. In: Procedia Computer Science, vol. 176 Elsevier B.V.; 2020. p. 3379–3388.
4. Bakar NH, Kasirun ZM, Salleh N. Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review. Journal of Systems and Software 2015 8;106:132–149.
5. Haris MS, Kurniawan TA, Ramdani F. Automated features extraction from software requirements specification (SRS) documents as the basis of software product line (SPL) engineering. Journal of Information Technology and Computer Science 2020;5:279–292. www.jitecs.ub.ac.id.
6. Putri DGP, Siahaan DO. Ekstraksi fitur produk perangkat lunak dari data opini pengguna. Jurnal ELTEK 2016;14:1–12.

7. Bakar NH, Kasirun ZM, Salleh N, Jalab HA. Extracting features from online software reviews to aid requirements reuse. Applied Soft Computing Journal 2016 12;49:1297–1315.

8. Siahaan D, Sarwosri, Azizah H. Ekstraksi Fitur Produk Perangkat Lunak dari Data Opini Pengguna. In: Konferensi Nasional Teknologi Informasi dan Aplikaisnya, vol. 2; 2016. p. F1–F6. http://seminar.ilkom.unsri.ac.id/index.php/kntia/article/view/721.

9. Haque MA, Rahman MA, Siddik MS. Non-Functional Requirements Classification with Feature Extraction and Machine Learning: An Empirical Study. In: 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019 (ICASERT 2019); 2019. p. 1–5.

10. Putri DGP, Siahaan DO. Software Feature Extraction Using Infrequent Feature Extraction. In: Proceedings - 2016 6th International Annual Engineering Seminar, InAES 2016; 2017. p. 165–169.

11. Htay SS, Lynn KT. Extracting product features and opinion words using pattern knowledge in customer reviews. The Scientific World Journal 2013;2013:1–5.

12. Bakar NH, Kasirun ZM, Salleh N. Terms Extractions: An Approach for Requirements Reuse. In: 2015 IEEE 2nd International Conference on InformationScience and Security, ICISS 2015; 2015. p. 1–4.

13. Raharjana IK, Aprillya V, Zaman B, Justitia A, Fauzi SSM. Enhancing software feature extraction results using sentiment analysis to aid requirements reuse. Computers 2021 3;10:36.

14. Hasrina N, Kasirun ZMB, Salleh N, Halim AH. Extracting Software Features From Online Reviews to Demonstrate Requirements Reuse in Software Engineering. In: 6th International Conference on Computing & Informatics, ICOCI 2017; 2017. p. 184–190.

How to cite this article: Manek P.G., Septiyanto A.F., Nugroho A.S. (2021), A Semantic Comparison of Feature Requirements Extraction Methods, *IPTEK The Journal of Technology and Science*, *32(3):176-183*.