

Goal Model Construction Based on User Review Classification

Hironori Shimada, Hiroyuki Nakagawa, and Tatsuhiro Tsuchiya
Graduate School of Information Science and Technology
Osaka University, Suita-shi, 565-0871, Japan

Abstract

In a software development process, developers should elicit various requirements correctly; however, it is difficult to capture all of the requirements. User reviews can be useful source to capture bug information or features that users request. In this paper, we propose an approach to construct a goal model from user reviews of an application. The goal model visualizes requirements in a hierarchical structure and helps the developer to obtain an overview of the requirements. To evaluate the proposed approach, we implement a tool that automates the approach and construct a goal model from mobile application reviews.

1 Introduction

Smart phones have spread widely and various mobile applications are developed, such as applications for SNS (Social Network Services), business, education, and game. To improve applications, developers should investigate user reviews and extract important requirements. However, since the number of reviews can be large, it is difficult to investigate all reviews to extract requirements correctly.

In this paper, we propose a systematic approach that constructs a goal model. In a goal model, goals, which represent the states that a system should accomplish, are organized in a tree form. The root goal, which the system has to finally accomplish, is decomposed into subgoals. To simplify the goal model construction, we regard required functions to be implemented as goals. We develop a prototype tool that implements the construction method. This tool can automatically construct a goal model. We evaluate whether the tool can correctly generate a goal model from reviews.

2 Background

Requirements analysis is an important process in software development. In this process, analysts organize requirements that are obtained from reviews or interviews. The analysts also have to analyze requirements conflicts and define countermeasures. Requirements incompleteness may cause insufficient implementation of functions that users request. To prevent requirement incompleteness, various requirements analysis models are proposed in prior work. In our research, we focus on the goal model, one of the requirements analysis models. The goal model visualizes requirements as goals in a hierarchical structure. It also enables the analysts to decompose goals into subgoals systematically and to find implicit goals easily. Examples of goal models include KAOS [2], i^* [9], NFR [6], and AGORA [5]. In our previous work, we proposed an approach of goal model elaboration for localizing changes in software evolution [7].

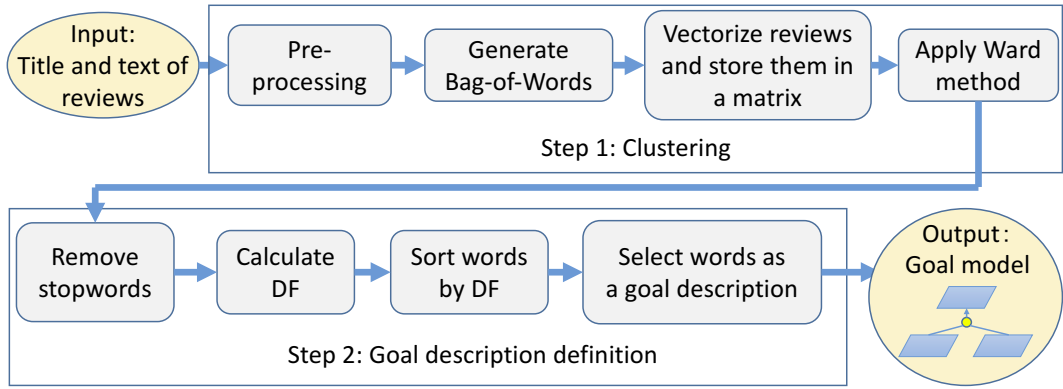


Figure 1: Overview of the goal model construction.

Some research focuses on collecting and using user reviews. Chen et al. [1] proposed a framework, AR-Miner, that shows grouped and ranked reviews to analysts. The framework filters non-informative reviews and groups reviews by topic modeling. Finally, it visualizes useful information and facilitates the developers to mine information. Fu et al. [3] examined the validity of each review by digitization of emotion that a word represents. They implemented a tool, WisCom, that discovers inconsistencies in reviews and identifies reasons why users like or dislike an application. Williams and Mahmoud [8] proposed a method of collecting and organizing technical information such as software bug reports and user requests from Twitter. In this paper, we propose an approach that visualizes requirements as goals that are derived from reviews.

3 Goal Model Construction

3.1 Overview

Fig. 1 illustrates the overview of our goal model construction method. The construction method is mainly composed of two steps: *clustering* and *goal definition*. This method uses a hierarchical clustering method to visualize application reviews in a hierarchical structure. The clustering method is appropriate for our purpose of aggregating reviews with respect to functions. In the hierarchical clustering method, clusters are coupled in order of the distance between clusters and embedded into a hierarchical tree structure. We interpret coupled clusters as goals in the goal model. In the next step, we label each coupled cluster with a goal description. This task is called goal definition. To convert the resulting clustering tree to a goal model, representative words are selected as a goal description from each coupled cluster. We explain the details of each step below.

Step 1: Clustering. Fig. 2 illustrates the overview of the clustering. Each review as input data contains its title and body text. First, reviews are pre-processed by lemmatization and stopwords elimination. Then a bag-of-words (BoW) is generated from each review and stored in a matrix as a vector. Finally, the Ward method [4], which is a hierarchical clustering method, is applied. The Ward method couples clusters in order of the distance between clusters and organizes them into a tree structure. We use the tree structure as the structure of the goal model to be constructed. Moreover, compared with non-hierarchical clustering methods such as k -means clustering, the hierarchical clustering method does not require the user to fix the number of clusters in advance. Since our approach aims to cluster reviews with respect to functions, it is difficult to predict the number of clusters. Therefore, we use the hierarchical clustering method for constructing a goal model.

Step 2: Goal definition. Fig. 3 illustrates the overview of the goal definition. In the previous step, pairs of clusters whose distance between them is close are coupled to generate a hierarchical structure. To construct a goal model from the clustering result, we regard the coupled clusters as goals. This step first weightens words in a cluster and then selects several words that characterize reviews in the cluster as a goal description. Before weightening words, general words that do not represent functions should be excluded. We use DF (document frequency), which represents the number of documents where the word appears, as the weight of a word. Several words are selected to define goal descriptions according to the DF values. Through these two steps, a goal model is constructed from user reviews. Fig. 4 illustrates an example of a generated goal model.

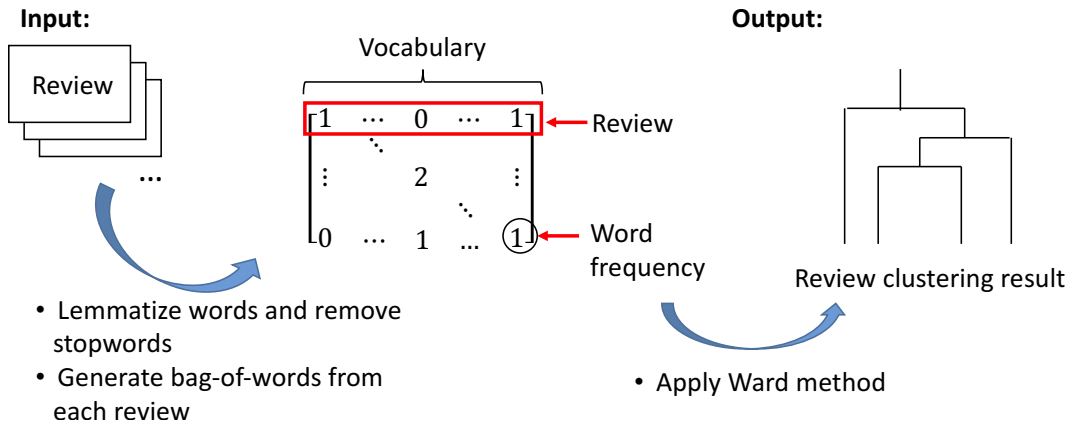


Figure 2: Overview of the clustering (step 1).

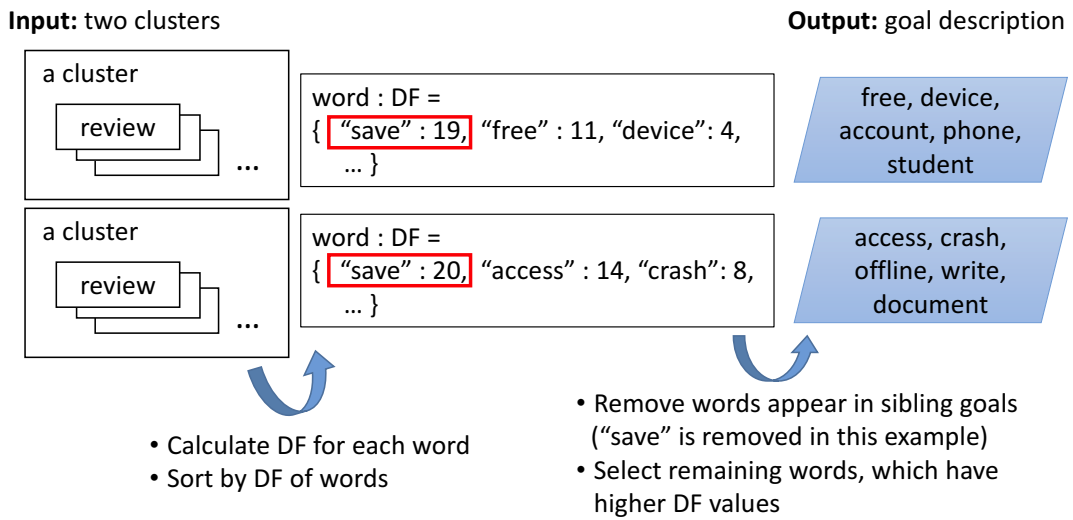


Figure 3: Overview of the goal definition (step 2).

4 Experiment and Evaluation

We examined whether the construction method can construct an appropriate goal model from user reviews. We implemented a tool that automatically performs the construction method. The Python programming language, and NLTK for word division and lemmatization, and Gensim library for generating bag-of-words are used for tool implementation.

Experimental Method. We performed an experiment as follows. First, we collected 500 reviews of Google Docs application from Apple App Store using iTunes search API. Google Docs is a file manager application to edit, save and share files. It also has a function to access files offline. In Google Docs, users can access files by the application using PC, mobile phone, or tablet. Next, we manually removed reviews that are not described in English. We finally used 491 reviews as the input of the tool. We also manually constructed a goal model from the application reviews for comparison. Finally, we constructed a goal model by the proposed approach. By tuning a design parameter, we let the automatically generate goal model have the same number of leaf goals as the manually constructed one for comparison purposes.

Experimental Results and Discussion. In the evaluation, we focus on leaf goals and evaluate how closely each leaf goal in the automatically generated goal model matches a leaf goal in the manually generated goal model. When constructing the goal models, leaf goal nodes are created by aggregating a set of reviews. Thus every review is associated with one of the leaf goals in both goal models. Now let $r_{i,j} = n_j/n_i$ where n_i is the number of reviews associated with leaf goal i in the automatically generated goal model and n_j is the number

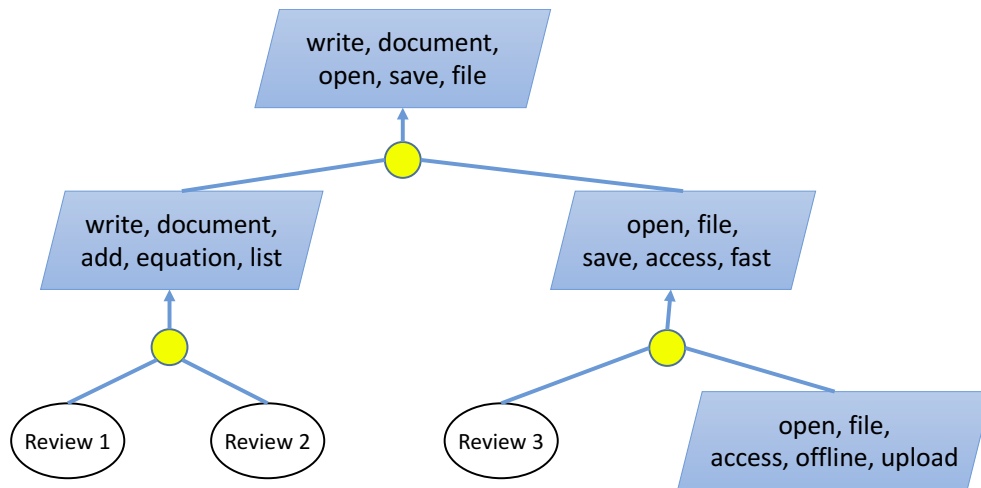


Figure 4: The example of a constructed goal model

of those in the n_i reviews that are associated with leaf goal j in the manually generated goal model.

Fig. 5 shows top three $r_{i,j}$ for each automatically generated leaf goal i . For example, let i be the third goal “phone, time, space, press, responsive.” This goal shares associated reviews with two goals in the manually constructed goal model. In fact, three reviews are associated with i and two of them are associated with one manually generated leaf goal j_1 , while the remaining one is associated with another leaf goal j_2 . Hence $r_{i,j_1} = 2/3$ and $r_{i,j_2} = 1/3$.

The results show that for many of the automatically generated leaf goals, a majority of their associated reviews are concentrated on one or two manually generated goals, which means that the leaf goals in the two goal models match each other to a substantial extent.

5 Conclusions

In this paper, we reported the experience of the goal model construction based on user review classification. We developed a prototype tool that implements the proposed approach. In the experiment, we constructed a goal model from user reviews and evaluated the results obtained.

For future work, we identified the following improvement points:

- **Goal description readability:** in our approach, words in reviews are weighted and several words are selected as a goal description. However, some goal descriptions are still difficult to understand the corresponding functions. To improve the goal description readability, we may have to redefine the metrics instead of DF.
- **The quality of a goal model:** the final objective of our research is to construct a goal model that explicitly visualizes function requests and bug information. To accomplish this objective, the tool should introduce a mechanism that visualizes goal types, such as function request or bug information.

References

- [1] Chen, N., Lin, J., Hoi, S.C.H., Xiao, X., Zhang, B.: Ar-miner: Mining informative reviews for developers from mobile app marketplace. In: Proceedings of the 36th International Conference on Software Engineering. pp. 767–778. ICSE 2014, ACM (2014), <http://doi.acm.org/10.1145/2568225.2568263>
- [2] Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. *Science of Computer Programming* **20**(1-2), 3–50 (Apr 1993)
- [3] Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J., Sadeh, N.: Why people hate your app: Making sense of user feedback in a mobile app store. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1276–1284. KDD '13, ACM (2013), <http://doi.acm.org/10.1145/2487575.2488202>

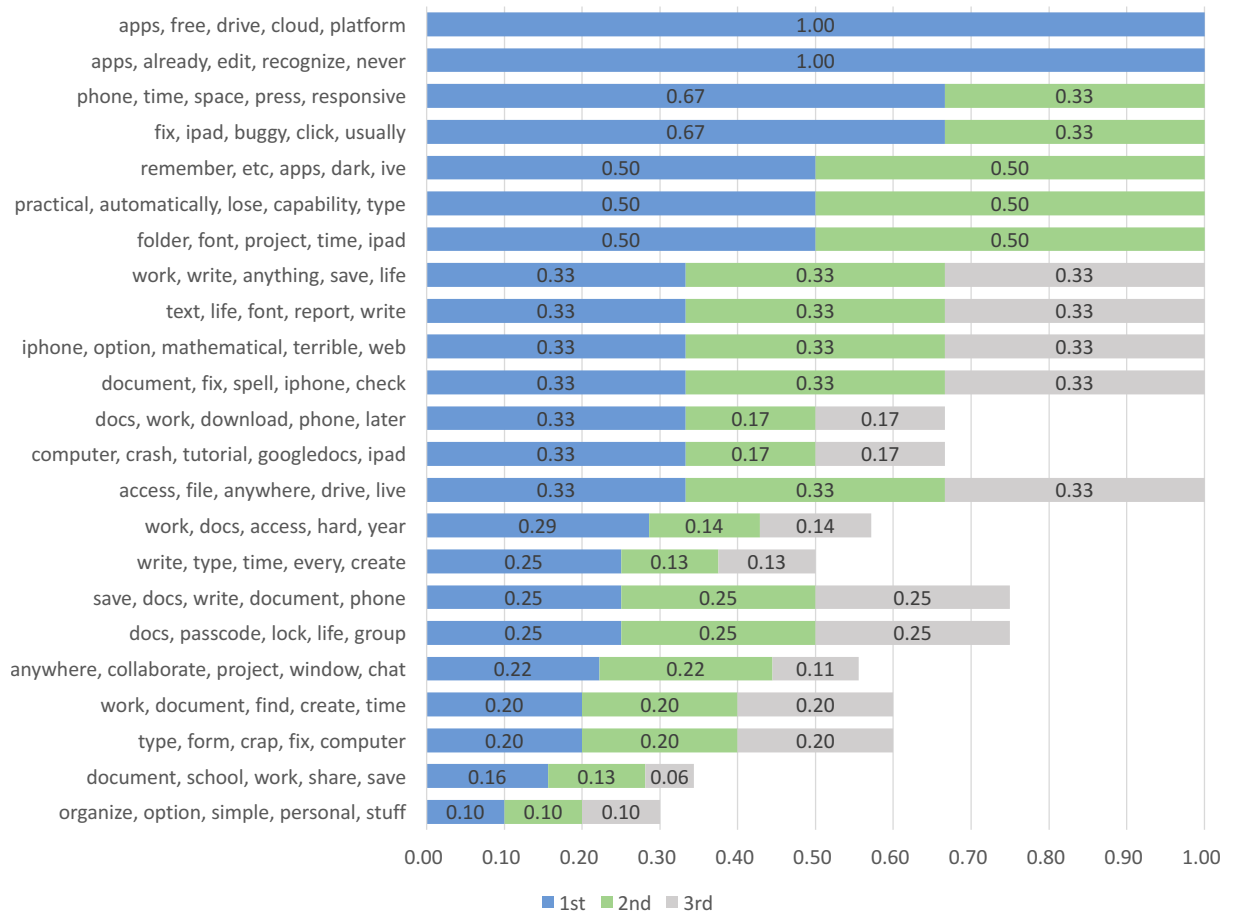


Figure 5: Aggregation rates of auto-generated goals.

- [4] Jr., J.H.W.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301), 236–244 (1963), <https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845>
- [5] Kaiya, H., Horai, H., Saeki, M.: Agora: attributed goal-oriented requirements analysis method. In: Proceedings IEEE Joint International Conference on Requirements Engineering. pp. 13–22 (2002)
- [6] Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: a process-oriented approach. IEEE Transactions on Software Engineering **18**(6), 483–497 (June 1992)
- [7] Nakagawa, H., Ohsuga, A., Honiden, S.: A goal model elaboration for localizing changes in software evolution. In: 2013 21st IEEE International Requirements Engineering Conference (RE). pp. 155–164 (July 2013). <https://doi.org/10.1109/RE.2013.6636715>
- [8] Williams, G., Mahmoud, A.: Mining twitter feeds for software user requirements. In: 2017 IEEE 25th International Requirements Engineering Conference (RE). pp. 1–10 (September 2017)
- [9] Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In: Proceedings of the Third IEEE International Symposium on Requirements Engineering, 1997. pp. 226–235 (January 1997)