4L-03

Supporting Feature Extraction from Natural Language Requirements Specifications

Yuedong Xiao, Kenji Hisazumi, and Akira Fukuda

Kyushu University

1 Introduction

Software product line (SPL) refers to software engineering methods, tools, and techniques for creating a collection of similar software systems from a shared set of software assets using a common means of production. [1] In software product line development, it is important to clearly grasp the things (commonality) common to all the products included in the product line and the things (variability) that can change for each product. Several approaches to explicitly model such commonality and variability have been proposed. According to the definition of Kang et al., [2] software feature is defined as a prominent or distinctive user-visible aspect, quality, or characteristic of a software system or systems. By selecting variable features based on product specifications in addition to common features that are intersections between products, individual mass production of product variants is possible. Recently, approaches that focus on requirements to recover variability information have been proposed because requirements contain more comprehensive information about commonalities and variability. [3]

The present work extends the use of the approach [4] for commonality and variability mining from domain-specific natural language documents. In addition, an effort is made to improve the accuracy of extracted features.

The remainder of this paper is structured as follows. In Section 2, related work is briefly discussed. Section 3 proposes an approach and introduces the particular techniques and how they are applied in context. Section 4 presents the result of a case study. Section 5 draws some concluding remarks and outlines future work.

2 Related Work

In Mining Commonalities and Variabilities from Natural Language Documents, Ferrari et al. (2013) [4] identified conceptually independent expressions (i.e., terms) through POS tagging, Linguistic Filters (filtering terms with adjectives and nouns), and lastly identifying C-NC value that computed term hood metric. Then, Contrastive Analysis was applied to select the terms that were domain-specific. If a term is domain-specific and appears in all of the documents, it is more likely to be categorized as a common feature. If a domain-specific term appears in some of the documents of the different vendors, but not in all documents, it is more likely to be considered as a variant feature. However, this approach exhibits a critical limitation: the accuracy which related to synonyms. Regardless of the technology, the main difference between [4] and our work is that we show care for synonyms.

3 Proposed Method

Overview In this section, we provide an overview of our approach and introduce the particular techniques and how they are applied in context. We provide an overview of the entire process in Figure 1. First, we select a requirements specification as our input file for preprocessing. Second, we identify a list of terms in an automatically POS-tagged text, which is then weighted with the C-value, currently considered as the state-of-the-art method for terminology extraction. Third, contrastive analysis is performed to revise the ranking of terms. Fourth, the list is used with a clustering algorithm to group terms which describe similar functionality into a cluster based on the word vectors. To compute the average ranking scores of each cluster, we can get a list of clusters.

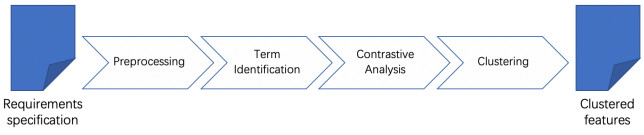

Figure 1: overview

Preprocessing Requirements specifications are very complex materials with both format and content. Code, formula, table, and figure are mixed in, which will interfere with our analysis of part of speech. Preprocess the text with regular expressions, retaining

only English characters, Japanese characters, and full-width punctuation marks. Since we have to ensure that the morphological analysis tool can accurately identify the part of speech. The following steps are then executed to identify features.

Terms Identification First, we use MeCab, a text segmentation library for use with text written in the Japanese language, to analyze and segment sentences into its parts of speech. Second, we select all those words or groups of words (referred in the following as multi-words) that follow a set of specific POS patterns (i.e., sequences of POS), that we consider relevant in our context. For example, we will not be interested in those multi-words that end with a postposition, while we are interested in multi-words with a format like:

$$(noun)^+ \tag{1}$$

$$(noun)^+ + particle + verb \tag{2}$$

Terms are finally identified and ranked by computing a “termhood” metric, called C-value. This metric establishes how much a multi-word is likely to be conceptually independent of the context in which it appears.

Contrastive Analysis Contrastive analysis is based on the assumption that a term that frequently occurs in the generic document is not likely to be a domain-specific term of requirements specification. With the contrastive analysis step, we take a generic contrastive document containing domain-generic terminology as input to extract a ranked list of terms with the same method described in Sect. 3.3. Then we compare these two lists of terms that if a term is less frequent in the contrastive corpora, it is considered as a domain-specific term, and it is ranked higher.

Clustering The previous step leads to a ranked list of terms where all the terms might be domain-specific. In the list, we notice an issue related to synonyms that a lot of terms represent the same feature. We apply Hierarchical Agglomerative Clustering (HAC) to improve similar terms identification. HAC is a method of cluster analysis which seeks to build a hierarchy of clusters. This algorithm repeat grouping the most similar groups and only one group remains until the end. Since word vectors are needed to be fed into HAC algorithm. We utilize word embedding (a Word2Vec model trained with Wikipedia corpus) instead of using traditional distributional semantic models to gain word vector representation of the requirements, which contains more semantic information. The results of hierarchical clustering are presented in a dendrogram. Finally, we set the numbers of clusters and compute the average ranking scores of each cluster. The more a cluster likely to contain features, the higher the ranking.

4 Case Study

We conducted a case study using the requirements specification on electric water boiler.[1] The size of the requirements specification text file is 30kB and it contains 9554 words. We selected some other home appliance manuals and some specification of other domains as contrastive materials. The Word2Vec model was trained on Japanese Wikipedia by Gensim and MeCab. Furthermore, the word vector of terms that did not exist in Wikipedia will be set to zero vector. We did an additional training to the model on abovementioned documents to prevent it.

A total of 257 terms were extracted from the requirements specification. Here are some examples of the term candidates with high ranking scores:”沸騰行為”,”保温行為”,”水位センサ”,”温度制御方式”,”ヒータ用電源”,”蓋センサ OFF”,”タイマボタン”,”水位メータ”,”ブザーを鳴らす”,”温度制御停止” etc.. We argue that a domain expert can easily recognize whether these terms can be regarded as features or not.

5 Conclusion

In this paper, we proposed an approach that focuses on dealing with synonyms by a HAC technique to support extracting features from natural language requirements specifications. The performance of the system is being evaluated through the experiment. We would like to verify the applicability of this approach in a real SPL context.

References

[1] Paul Clements and Linda Northrop. *Software product lines: practices and patterns*, volume 3. Addison-Wesley Reading, 2002.

[2] Kyo C Kang, Sholom G Cohen, James A Hess, William E Novak, and A Spencer Peterson. Feature-oriented domain analysis (foda) feasibility study. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[3] Noor Hasrina Bakar, Zarinah M Kasirun, and Norsaremah Salleh. Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review. *Journal of Systems and Software*, 106:132–149, 2015.

[4] Alessio Ferrari, Giorgio O Spagnolo, and Felice Dell’Orletta. Mining commonalities and variabilities from natural language documents. In *Proceedings of the 17th International Software Product Line Conference*, pages 116–120. ACM, 2013.

[1] 話題沸騰ポット (GOMA-105 型) 要求仕様書