Automated Requirement Sentences Extraction from Software Requirement Specification Document

M Syauqi Haris  
Faculty of Computer Science  
Brawijaya University  
Malang Indonesia  
syauqi@student.ub.ac.id

Tri Astoto Kurniawan  
Faculty of Computer Science  
Brawijaya University  
Malang Indonesia  
triak@ub.ac.id

ABSTRACT
In the requirement reuse and natural language document-based Software Product Line (SPL) domain analysis, requirement sentences of the requirement document are the primary concern. Most studies conducted in this research area have document preprocessing stage in their methods that is a manual process to separate requirement sentences and non-requirement sentences from the document. This manual labor process might be tedious and error-prone since it will need much time and expert intervention to make this process completely done. In this paper, we present a method to automate requirement sentence extraction from the Software Requirement Specification (SRS) document by leveraging Natural Language Processing (NLP) approach and requirement boilerplate sentence patterns. Conducted experiments in this research show this method has such accuracy from 64% to 100% on precision value and recall value in the range of 64% to 89%.

CCS CONCEPTS
• Software and its engineering ~ Software notation and tools ~ System description language ~ Specification language.

KEYWORDS
Natural language processing, Software requirement reuse, Requirement boilerplate, Software product line, Domain analysis.

1. INTRODUCTION
Requirement sentence extraction from the SRS document is needed for studies on software features information retrieval as the basis of software product line domain analysis [1][2][3]. Although the software features extraction process is automated in those studies, but the requirement sentences are previously prepared by the manual work of an expert. It is also found in the study on ambiguous software requirement specification detection [4]. As such, these studies represent many other studies that still need expert intervention to manually process the SRS document to extract requirement sentences that might be tedious and also error-prone.

Natural Language Processing (NLP) approach is widely used for information retrieval automation from natural language documents [5][6][7]. NLP pipelines such as tokenizer, sentencer, POS tagger, Named Entity Recognizer (NER), and chunker are used for automated extraction and clustering of requirements glossary term [6]. While NLP tokenization and Part-of-Speech tagging are the most used NLP technique for text pre-processing [7]. Based on these studies, it is possible to automate information extraction from the SRS document by understanding the patterns of how the information itself is written in the document.

Requirement boilerplate is a standardized way to express software requirements in natural language [8]. The use of requirement boilerplate encourages the practice of writing concise and consistent requirement sentences in the document. Ambiguous software specification also can be detected early if the requirement is stated in the boilerplate template [4]. Based on this concept, by leveraging requirement boilerplate syntax, requirement sentence patterns can be formulated as the basis of the automated extraction process.

This research will propose a method for the requirement sentence extraction process from the SRS document directly as a computer file without any manual work intervention. Requirement sentence expressed in requirement boilerplate pattern will be identified by NLP Part-of-Speech (POS) tag sequence based on the several proposed rules that are constructed based on the SRS dataset.

This paper is constructed as follows. Section 2 will situate related works that were previously conducted to show state of the art in this research area. Section 3 will present the basic
2. RELATED WORKS

Osman et al. [4] proposed a method to automatically detect ambiguous software requirement specification by evaluating nine classification algorithms and choose Random Forest as the best performing classification model. While pre-processing SRS document activity such as requirement sentence extraction, word tagging, and labeling is done manually by an expert.

Nan Niu et al. [9] defined the functional requirements profile (FRP) resulting from a ‘verb-direct object’ relation extraction from SRS functional requirement sentences in the requirements natural language document. Extracted FRPs are then processed with orthogonal variability modeling (OVM) to determine the variabilities and the constraints for extractive software product line (SPL) domain analysis.

Arora et al. [10][11] introduced an automated method for requirement glossary terms extraction from natural language software requirement documentation. NLP syntactic parsing approach combined with clustering algorithms such as K-means, Hierarchical Agglomerative, and Expectation Maximization is used to extract terms candidate and selecting them from the SRS document.

Most studies on information extraction from natural language documents such as SRS are motivated by the extractive software product line concept, where the domain analysis is based on the extracted features from existing systems. Even though the domain engineering process is identical with feature extraction instead of requirement sentence extraction, but feature here means an abstraction from functional requirements. Therefore, requirement sentence extraction still an important part of the studies.

3. BASIC NOTIONS

3.1. Software Requirements Specification (SRS)

Software Requirements Specification (SRS) document is the official document of what the system developers should implement. Since this document is made for various stakeholders, it may include both the user requirements for a system and a detailed specification of the system requirements. The level of document detail depends on the type of system developed and the development process used. The critical system will need more detailed documentation than the non-critical system. As well as when the development process is done by a separate company or outsourced, the document needs to be more detailed and concise than an in-house and iterative development process is used [12].

Several large organizations, such as the U.S. Department of Defense and the IEEE, have defined standards for requirements documents. These are usually very generic but are nevertheless useful as a basis for developing more detailed organizational standards. The U.S. Institute of Electrical and Electronics Engineers (IEEE) is one of the best-known standards providers, and they have developed a standard for the structure of requirements documents [13]. This standard is most appropriate for systems such as military command and control systems that have a long lifetime and are usually developed by a group of organizations.

In this study, we only select SRS documents that use requirement boilerplate to represent requirement sentences. SRS documents are not limited to certain standardized formats such as IEEE 830-1998.

3.2. Requirement Boilerplate

In writing user requirements, the statement of functional requirements and non-functional requirements must be understood by system users who may not have technical knowledge. Therefore, the use of technical jargon and formal language notation must be avoided. On the contrary, what should be done is the use of natural language accompanied by explanatory pictures, simple tables, and diagrams that are easy to understand [12]. Consistent use of language will make a requirement statement easy to understand and identify. A simple example is the use of the word “shall,” which indicates the existence of a statement of need in a sentence. Another example is the choice of the words “shall”, “should”, and “may”, which express different levels of priority for a requirement [8].

Requirements boilerplate is a template that provides a simple and understandable approach to reduce language effects when documenting requirements. This guidance will support the software requirement author in achieving high quality and syntactic unambiguously sentences in optimal time and at low costs [14]. An illustration of the requirement boilerplate template is shown in Figure 1.

![Figure 1: Requirement Boilerplate Template [14]](image-url)
3.3. Natural Language Processing (NLP)

Currently, NLP is widely used in researches working on human language documents, since there are many public NLP libraries to use, i.e. Google SyntaxNet, Stanford CoreNLP, NLTK Phyton library, and spaCy [15]. NLP library provides several pipelines to process the documents as the input are illustrated in Figure 2. Sentencizer can split text document to sentences series, Tokenizer can split document or sentence to token series, token itself can be an individual word or words phrase when the text chunking method is applied, Part-of-Speech (POS) tag is code tagging for every token identified, it can be noun, verb, determiner, adjective, adverb, etc. [16]. NLP library also can have a pre-trained model to add more capability on Named Entity Recognizer (NER) to automatically identify detected entity as person, organization, nation, etc. It also can have a Dependency Parser for text chunking (phrase detection) or sentence boundary detection. Both NER and dependency parser is available in the spaCy NLP library [17]. The illustration of these NLP pipelines is shown in Figure 2.

![Figure 2: NLP Pipelines for Information Extraction from Natural Language Document [18]](image)

This research did not use all the pipelines available in the NLP library but only sentence segmentation (sentencizer), tokenization, and part-of-speech tagging. This method is chosen because reducing pipelines loaded on processing means optimizing and time-saving in spaCy1.

4. METHODOLOGY

4.1. Research Question

In the context of the research area specified in section 1, this paper intended to answer the following research questions:

**RQ1.** What approach or technique needs to be constructed to automatically extract requirement sentences directly from the SRS document?

**RQ2.** What are the patterns of requirement sentences in the NLP concept which allow such formulation on certain rules that can be applied to automate requirement sentence extraction from the SRS document?

**RQ3.** How accurate the formulated rules to identify all requirement sentences that exist along in the whole SRS document?

4.2. Data Acquisition

SRS documents used in this study are chosen from Public Requirement Engineering (PURE)2 dataset [19]. Since this research focused on automated requirement sentence extraction from the SRS document, so this PURE dataset is appropriate to be used. Dataset rationalization is also made to choose certain SRS documents that are viable for manual checking described in Figure 3.

![Figure 3: Dataset Rationalization Illustration](image)

From 79 SRS documents in the dataset, we have done manual processing to find the part of the document that contains the functional requirements and observe whether each sentence of the requirement is presented using boilerplate requirements or not. We found 13 documents consistently using requirement boilerplate to express functional requirement sentence.

Finally, this research only selects 5 SRS documents that have less than 500 sentences and have consistent system naming based on the requirement boilerplate template. This consideration is made because we assumed that a document with too many sentences would make the evaluation process difficult since the correctness and completeness assessment is done manually. Five selected SRS documents are presented in Table 1.

<table>
<thead>
<tr>
<th>No</th>
<th>SRS Document Title</th>
<th>Number of Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Crime &amp; Criminal Tracking Network and Systems (CCTNS)</td>
<td>431</td>
</tr>
<tr>
<td>2</td>
<td>Digital Home System (DH)</td>
<td>404</td>
</tr>
<tr>
<td>3</td>
<td>e-Store</td>
<td>322</td>
</tr>
<tr>
<td>4</td>
<td>Laboratory Information System (LIS)</td>
<td>314</td>
</tr>
<tr>
<td>5</td>
<td>Puget Sound Enhancements (Puget)</td>
<td>495</td>
</tr>
</tbody>
</table>

4.3 Solution Development

Processing natural language documents such as the SRS document will involve NLP pipelines on processing the document. From all NLP libraries that are widely available for the various programming language, spaCy, the one with Python

1 https://spacy.io/usage/processing-pipelines

2 http://nlreqdataset.isti.cnr.it
language compatibility and has a pre-trained model, is chosen for this research.

In the document processing with NLP, NLP pipelines are used to facilitate in acquiring the language properties. Utilizing language properties will need some rules following the information that will be retrieved, which is requirement sentences in this research context. Therefore, by analyzing the selected SRS document from the dataset, several rules are then proposed to identify a particular sentence that represents requirement boilerplate syntax implementation in the document. These data-driven rules are then applied to software automation tools to extract as much as possible requirement sentences that exist in the SRS document.

Text pre-processing means the early step to prepare the raw text data that was extracted from the computer file, so the text can be processed further by NLP. Since the main focus of the extraction objects are the requirement sentences, pre-processing in this research has an intention to reduce complexity and simplify the requirement sentence variation in the document. The rules proposed are also based on the variations found in the selected dataset.

4.4 Evaluation Metrics

Since this research focuses on automating the manual work for extracting requirement sentences from the SRS document, the accuracy metric will be calculated by comparing the result from the automation process to the actual result by manual extraction. Precision and recall will be used to measure how accurate the solution accuracy. Precision value is the ratio between the true result acquired or true positive (TP) to all the relevant elements that should be acquired or all positive (TP + FP), as shown in Formula (1). While the recall value is the ratio between the true result acquired (TP) to all the acquired elements, whether it is true or false (TP + FN), as shown in Formula (2).

\[
\text{precision} = \frac{TP}{TP + FP} \tag{1}
\]

\[
\text{recall} = \frac{TP}{TP + FN} \tag{2}
\]

5 PROPOSED SOLUTION

5.3 Solution Framework

The requirement sentence extraction automation framework from SRS documents is the basic conceptual structure in this study. In this framework, the SRS document is extracted to produce raw text, and then the raw text is pre-processed to eliminate certain sentence variations noises and ready to be processed with NLP. NLP in this framework uses the SpaCy NLP library with a pre-trained model. The first process in NLP is sentencizer and tokenizer to break up pre-processed text documents into sentences consisting of tokens (separated words), while noun-chunking is used to recognize phrases so that they are not considered as separated words. The next process is implementing the POS tagging for every token and then applying the POS tag sequence rule to automatically identify requirement sentences. An overview of the requirement sentence extraction automation framework from the SRS document is presented in Figure 4.

![Figure 4: Requirement Sentences Extraction Automation Framework](image)

5.4 Text Pre-Processing Rules

After extracting text data from the SRS document, there are several preliminary processes for the text data before it can be processed by the NLP engine. In this process, variations in requirement boilerplate sentences are simplified to facilitate the requirement sentence keyword recognition for further NLP processing. In this stage, these rules are applied to all sentences in the text data. Sentence simplification rules are illustrated in Figure 5.

![Figure 5: Sentence Simplification Rules Illustration](image)
5.5 Part-of-Speech Tag Sequence Rules

Requirement sentences written in requirement boilerplate template has a pattern of specific word sequence, which can be leveraged to determine rules for requirement sentence identification. This research used 2 (two) POS tag modifications, SYSNOUN as a system word marker or phrase containing the system word, and SYSVERB as a keyword marker for the keyword of requirement boilerplate, i.e., the verb “shall”. The illustration of the POS tagging result is presented in Figure 6.

![Figure 6: Modified Part-of-Speech (POS) Tagging Illustration](image)

Based on 5 (five) SRS documents selected from the PURE dataset, Crime & Criminal Tracking Network and Systems (CCTNS), Digital Home System (DH), e-Store, Laboratory Information System (LIS), and Puget Sound Enhancements (Puget), existing structure in requirement sentences of each document make it possible to formulate 5 (five) POS tagging sequence rules as follows:

**Rule 1.** SYSNOUN-KEYVERB-VERB-NOUN

This rule is used to identify requirement sentences structure as follows:
- The system shall maintain the audit trail for as long as required. [CCTNS]
- The DH system shall backup all system data configuration default parameter settings planning and usage data. [DH]
- The system shall authenticate user credentials to view the profile. [e-Store]
- The system shall log an error message to the external data file. [LIS]
- The system shall display a search box on every page after an actor has logged in. [Puget]

**Rule 2.** SYSNOUN-KEYVERB-ADV-VERB-ADP-NOUN

This rule is used to identify requirement sentences structure as follows:
- The system shall run on multiple browsers. [CCTNS]
- The system shall adhere to the standards policies and procedures of the American society. [DH]
- The e-store system shall communicate to credit management system. [e-Store]
- The system shall clear up data if the user chooses to click the cancel button. [LIS]

**Rule 3.** SYSNOUN-KEYVERB-ADV-VERB-NOUN

This rule is used to identify requirement sentences structure as follows:
- The system shall optionally allow user to print the invoice [e-Store]

**Rule 4.** SYSNOUN-KEYVERB-ADV-VERB-ADP-NOUN

This rule is used to identify requirement sentences structure as follows:
- The system shall never include in the search result list any record which the user does not have the right to access. [CCTNS]
- The system shall automatically log out all customers after a period of inactivity. [e-Store]

**Rule 5.** SYSNOUN-KEYVERB-PART-VERB-NOUN

This rule is used to identify requirement sentences structure as follows:
- The system shall not include such cases in any count of search results, this level of security is normally appropriate for cases dealing with matters such as national security. [CCTNS]
- the system shall not leave any cookies on the customer’s computer containing any of the user’s confidential information. [e-Store]

In this research, we limit the identification only for system perspective requirement sentences, while the user perspective is excluded at the moment. Passive form sentences are also excluded since we follow the requirement boilerplates syntax guide that shown active form sentences only.

6 RESULT AND DISCUSSION

After implementing POS tag sequence rules on SRS documents, this research automatically produces the list of requirement sentences that are compared with actual requirement sentences (RS) from manual extraction. Precision and recall are computed afterward based on the number of results that are True Positive (TP), False Positive (FP), and False Negative (FN) from each SRS document processing. The comparison and accuracy calculations are presented in Table 2.

<table>
<thead>
<tr>
<th>Table 2: Requirement Sentence Extraction Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>CCTNS</td>
</tr>
<tr>
<td>DH</td>
</tr>
<tr>
<td>e-Store</td>
</tr>
<tr>
<td>LIS</td>
</tr>
<tr>
<td>Puget</td>
</tr>
</tbody>
</table>

The findings of this study can be interpreted from the requirement sentence extraction result comparison in Table 2. This study proves the highest precision when applied to the e-Store SRS document with 100% precision value. It means that the proposed method, when it is applied to the e-Store SRS document, did not fail to exclude non-requirement sentences in the extraction process. In the same document, this study shows the highest recall value of 89%. It means that the proposed method for the e-Store SRS document succeeded in extracting 89% of all the requirement sentences that should be obtained.
This study also found 64% precision value of Digital Home (DH) SRS document that shows 36% of false-positive (FP) extraction results and 64% recall value of Crime & Criminal Tracking Network and Systems (CCTNS) SRS document that shows 36% of false-negative (FN) extraction results. The cause of this extraction’s failures can be identified to see the limitation of the proposed method. The examples of these findings are presented in Table 3.

### Table 3: Failed Extraction Result Analysis

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Result</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system owner director shall provide management and communication support</td>
<td>FP</td>
<td>Using system phrase and boilerplate structure</td>
</tr>
<tr>
<td>The development of the dh system shall use methods and techniques such as ...</td>
<td>FP</td>
<td>Using system phrase and boilerplate structure</td>
</tr>
<tr>
<td>The user interfaces of the system shall comply with standard iso 9241</td>
<td>FP</td>
<td>Using system phrase and boilerplate structure</td>
</tr>
<tr>
<td>The System must not exceed &lt;xx&gt; hours per &lt;rolling three months period&gt;</td>
<td>FN</td>
<td>Using of symbol that not recognized</td>
</tr>
<tr>
<td>The System must be able to capture and store violations</td>
<td>FN</td>
<td>Using conjunction of two process verbs</td>
</tr>
<tr>
<td>The system should be developed to be deployed in a 3-tier datacenter architecture</td>
<td>FN</td>
<td>Using a passive sentence structure</td>
</tr>
</tbody>
</table>

### 7 CONCLUSION

This study gives a logical answer for all the research questions stated in section 4. The first research question on approach or technique is answered by the proposed solution framework in sub-section 5.2. The second research question about requirement sentence pattern is answered by part-of-speech tag sequence rules presented in sub-section 5.4. and the third research question on accuracy is answered in the result and discussion section in this paper, which is shown accuracy measurement on automated requirement sentence extraction from SRS document varies with precision value in the range of 64% to 100% and recall value in the range 64% to 89%. For the next study, an improvement in extraction rules or text pre-processing is needed to resolve false detection causes found that are shown in the discussion section.

### REFERENCES


