

## A Comparative Study on Requirements Analysis Techniques using Natural Language Processing and Machine Learning

Byung-Sun Cho\*, Seok-Won Lee\*\*

\*Student, Dept. of Computer Engineering, Ajou University, Suwon, Korea

\*\*Professor, Dept. of Software and Computer Engineering, Ajou University, Suwon, Korea

### [Abstract]

In this paper, we propose the methodology based on data-driven approach using Natural Language Processing and Machine Learning for classifying requirements into functional requirements and non-functional requirements. Through the analysis of the results of the requirements classification, we have learned that the trained models derived from requirements classification with data-preprocessing and classification algorithm based on the characteristics and information of existing requirements that used term weights based on TF and IDF outperformed the results that used stemming and stop words to classify the requirements into functional and non-functional requirements. This observation also shows that the term weight calculated without removal of the stemming and stop words influenced the results positively. Furthermore, we investigate an optimized method for the study of classifying software requirements into functional and non-functional requirements.

▶ **Key words:** Software Engineering, Requirements Engineering, Requirements Analysis, Machine Learning, Natural Language Processing

### [요 약]

본 연구의 목적은 다양한 도메인에 대한 소프트웨어 요구사항 명세서로부터 수집된 요구사항을 데이터로 활용하여 데이터 중심적 접근법(Data-driven Approach)의 연구를 통해 요구사항을 분류한다. 이 과정에서 기존 요구사항의 특징과 정보를 바탕으로 다양한 자연어처리를 이용한 데이터 전처리와 기계학습 모델을 통해 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류하고 각 조합의 결과를 제시한다. 그 결과로, 요구사항을 분류하는 과정에서, 자연어처리를 이용한 데이터 전처리에서는 어간 추출과 불용어제거와 같은 토큰의 개수와 종류를 감소하여 데이터의 희소성을 좀 더 밀집형태로 변형하는 데이터 전처리보다는 단어 빈도수와 역문서 빈도수를 기반으로 단어의 가중치를 계산하는 데이터 전처리가 다른 전처리보다 좋은 결과를 도출할 수 있었다. 이를 통해, 모든 단어를 고려하여 가중치 값은 기계학습에서 긍정적인 요인을 볼 수 있고 오히려 문장에서 의미 없는 단어를 제거하는 불용어 제거는 부정적인 요소로 확인할 수 있었다.

▶ **주제어:** 소프트웨어공학, 요구공학, 요구사항 분석, 기계학습, 자연어처리

- 
- First Author: Byung-Sun Cho, Corresponding Author: Seok-Won Lee
  - \*Byung-Sun Cho (chobs1028@ajou.ac.kr), Dept. of Computer Engineering, Ajou University
  - \*\*Seok-Won Lee (leesw@ajou.ac.kr), Dept. of Software and Computer Engineering, Ajou University
  - Received: 2020. 02. 10, Revised: 2020. 06. 17, Accepted: 2020. 06. 19.

## I. Introduction

### 1.1 Background

급변하는 비즈니스와 인간 생활의 복잡성으로 인하여 다양하게 요구되는 서비스에 신속하게 적응하고 대처하는 소프트웨어 또는 시스템 설계와 개발은 쉽지 않다. 또한, 불확실하고 빠르게 변해가는 환경 속에서 특정 도메인의 소프트웨어 또는 시스템은 신뢰성과 안정성에 대한 요구는 급속히 증가하고 있다. 과거와는 달리 소프트웨어와 시스템의 중요성이 증가하면서 복잡성, 확장성, 규모 등이 증가되는 동시에 소프트웨어와 시스템 개발에서 많은 요구사항들이 도출된다. 이와 같은 환경 속에서 요구사항 분석(Requirements Analysis)은 많은 프로젝트에서 필수적으로 요구 된다.[1]

소프트웨어공학(Software Engineering)에서 요구사항 분석(Requirements Analysis)은 소프트웨어에 대해 요구가 무엇인지를 추출하여 소프트웨어 요구사항 명세서를 작성하는 작업으로, 사용자의 요구를 추출하여 목표를 정하고 어떤 방식으로 해결할 것인지를 결정하는 단계이다.[2] 이를 통해, 구현될 시스템의 기능이나 목표, 제약사항 등과 사용자로부터 시스템 개발에 필요한 정보를 추출할 수 있다. 소프트웨어 또는 시스템 개발 초기에 요구되는 중요한 프로세스이며 소프트웨어의 모형 또는 개발 방향을 결정하는 첫 단계이다. 이와 같은 이유로, 소프트웨어 개발 주기(Software Development Life Cycle, SDLC)에서 큰 비중을 차지하고, 소프트웨어 개발 및 프로젝트의 성공 또는 실패를 구분 짓는 결정적인 요인이다.[3]

실제 대다수 소프트웨어 개발에 실패하는 요인으로는 의사소통의 실패로 이해관계자의 요구사항을 정확히 도출하지 못한 경우, 요구사항의 의미를 서로 다르게 이해하는 경우, 그리고 소프트웨어가 요구사항을 제대로 충족하지 못한 경우 등이 있다. 실제로 소프트웨어 요구사항은 소프트웨어 개발 분야에서 사용자의 필요성, 계약상에 명시된 내용, 표준 사양서, 작성된 공식 문서 등에서 찾아낸 소프트웨어가 반드시 수행해야 할 필요한 조건, 기능 또는 능력을 서술하고 있고 이러한 요구사항을 정확하게 분석하는 작업이 소프트웨어 개발에 중요성이 점차 커지면서 요구공학(Requirements Engineering)에서는 체계적인 분석 모델을 통해 요구사항 추출, 분석, 명세, 검증과 유지보수 및 관리하여 명확하게 요구사항을 분석한다.

요구공학에서 정의한 소프트웨어 요구사항에는 대표적으로 기능적 요구사항(Functional Requirements), 비기능적 요구사항(Non-functional Requirements) 그리고

도메인 요구사항(Domain Requirements)으로 구성되어 있다. 요구공학에서 소프트웨어 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류하는 일은 요구사항 분석 단계에서 중요한 작업 중 하나이다. 기능적 요구사항과 비기능적 요구사항의 정확한 분류는 소프트웨어 개발에서 정확한 요구사항 분석을 도출할 수 있으며, 더 나아가 특정 도메인에 대해서 소프트웨어의 신뢰성, 안정성, 보안성, 효율성 등과 같은 다양한 비기능적 소프트웨어 품질을 만족할 수 있는 믿을 수 있는 소프트웨어(Dependable Software)를 개발할 수 있다. 이를 통해, 소프트웨어의 기능적 요구사항과 비기능적 요구사항을 정확히 정의하고 신뢰성과 안정성을 가진 소프트웨어 개발을 통해 프로젝트의 실패를 피할 수 있다. 이러한 중요성 속에서 요구공학 분야에서는 요구사항을 기능적 요구사항과 비기능적 요구사항으로 정확한 분류에 대한 연구가 진행되고 있다.[4][5]

최근, 기계학습(Machine Learning)과 심층학습(Deep Learning)을 이용한 인공지능 연구 분야가 활발히 진행되는 동시에 기술의 발달로 인해 발생된 새로운 환경 속에서 다양한 시스템들은 자동화되고 있다. 많은 연구 및 산업 환경에서 이와 같은 기술들을 각 분야의 연구 문제들의 새로운 해결 방안, 시스템 또는 기술로 도입되어 전문가 시스템과 같은 의사 결정을 돕는 보조 형태 또는 대체하는 형태로 점차 발전되고 있다. 이는 요구공학 분야에서도 인공지능의 기계학습과 자연어처리를 도입하게 되는 계기가 되었고 오랜 시간동안 전문가에 의해 소프트웨어 요구사항에 대한 도출, 분석, 명세, 검증 과정들 또한 기계학습을 이용하여 보다 정확한 자동화 시스템에 대한 연구가 진행되고 있다.

### 1.2 Research Problem

오늘 날에는 많은 소프트웨어 요구사항 명세서(Software Requirements Specification, SRS)에서 명확하게 정의되고 묘사된 다양한 요구사항이 명세 되어있고 또한 쉽게 찾아볼 수 있다. 하지만 요구사항 명세서에 기능적 요구사항과 비기능적 요구사항으로 분류하는데 기존 시스템에서 많은 한계점을 보유하고 있다.[6] 먼저, 수동식 시스템(Manual System)에서는 소프트웨어공학 또는 요구공학 전문가가 요구사항 분석과정에서 직접 요구사항을 분석하고 분류한다. 이는 전문가로부터 많은 시간과 노력이 요구되며 규모가 크고 복잡한 소프트웨어의 많은 수의 요구사항을 직접 사람이 분석하고 분류하기에는 너무 어렵고 시간이 많이 소모되며 잦은 오류가 발생하기 쉬운 단점이 있다. 그래서 자동화 시스템(Automated System)은 이러한 수동식 시스템 방식의 한계점을 보완하고 해결하려고

했지만 새로운 문제점들이 발생하였다. 실제 요구사항은 자연어로 작성되어 있고 자연어 특성 때문에 그에 따른 문제들이 발견되었다. 자연어의 특성상 이해관계자들 간에 다른 용어의 사용, 동일한 요구사항의 서로 다른 묘사 및 해석 등은 자동화 분류 시스템이 자연어를 정확히 이해하고 분류하는데 있어서 많은 문제점을 초래한다. 그로 인해 자동화 분류 시스템을 통해 정확한 기능적 요구사항과 비기능적 요구사항의 분류 결과를 도출할 수 없다. [7][8]

### 1.3 Research Objective

본 연구의 목적은 다양한 도메인에 대한 소프트웨어 요구사항 명세서로부터 수집된 요구사항을 데이터로 활용하여 데이터 중심적 접근법의 연구를 통해 요구사항을 분류한다. 이 과정에서 기존 요구사항의 특징과 정보를 바탕으로 다양한 자연어처리를 이용한 데이터 전처리와 기계학습 모델을 통해 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류하고 각 조합의 결과를 제시한다. 이는, 자연어 처리와 기계학습을 이용한 도출된 객관적이고 일관성 있는 요구사항 분류 결과이며 새로운 요구사항 데이터에 대해 본 연구에서 도출된 결과를 기반으로 적합한 자연어처리와 기계학습 모델을 제안한다. 이를 통해, 소프트웨어 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류하는 연구에 대해 최적화된 방법을 제시한다. 더 나아가, 본 연구에서 도출된 결과를 통해 요구공학 전문가를 보조할 수 있고, 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류에 대해 효율적인 분석을 가능하게 할 수 있다.

## II. Related Works

Abad et al.[9]의 연구는 언어학적 지식 기반의 접근법을 통해 요구사항 분류의 연구를 진행하였다. 이를 통해, 문법적, 시간적, 그리고 감정적인 요소가 기계학습을 이용한 요구사항 분류에서 얼마나 영향을 미치는지 연구하였다. 이를 위해 데이터 전처리 과정에서 Hussain et al. [10]의 연구에서 제안한 특징 추출을 기반으로 특징 선택을 제안하여 요구사항을 기능적 요구사항과 비기능적 요구사항 분류에 대한 연구에 대해 여러 알고리즘을 적용하여 비교 연구하였다. 이 연구에서 기여도는 첫째로, 문법적, 시간적 그리고 감정적인 요소를 고려하여 요구사항 분류에 더 적합한 데이터 전처리 방법을 제안하고 이를 통해 분류 성능을 향상하였다. 두번째, 제안한 특징 선택을 이용하여 다양한 기계학습 알고리즘 적용 및 성능을 비교 분석하였다.

Dekhtyar et al.[11]의 연구는 워드 임베딩과 심층학습을 이용한 접근법을 통해 요구사항 분류의 연구를 진행하였다. Google의 word2vec와 Kim.[12]의 제안한 문서 분류의 합성곱 신경망을 통해 요구사항을 얼마나 더 정확히 분류할 수 있는지 연구하였다. 이 연구에서 기여도는 첫째로, Google의 word2vec를 통해 미리 사전 학습된 워드 임베딩을 통해 요구사항 분류 모델의 성능을 향상하였다. 두번째, 합성곱 신경망의 학습 파라미터를 다양하게 설정하여 요구사항 분류에 최적화된 환경에 대해 결과를 도출하였다.

본 연구에서는 관련 연구에서 부족했던 요구사항 데이터를 다른 요구사항 명세서로부터 추출 및 추가하였고, 다양한 자연어처리와 지도학습 알고리즘을 이용한 요구사항 데이터 전처리와 분류를 비교 분석하여 최적의 모델과 전처리 방법을 찾는 것은 중요하다.

## III. The Proposed Method

### 3.1 Proposed process model

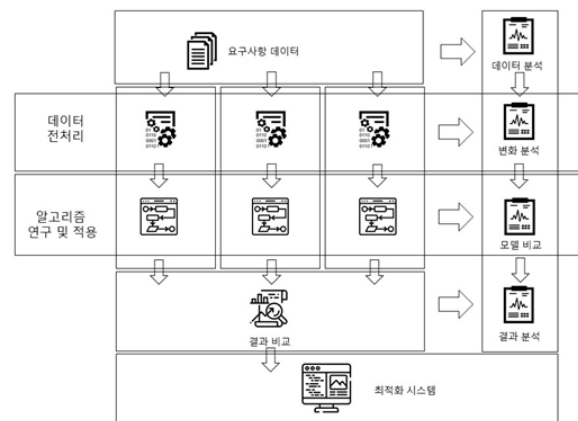


Fig. 1. Proposed process model

먼저, 기존에 요구 공학에서 공개한 요구사항 데이터셋을 제외하고 다른 연구에서 사용된 소프트웨어 요구사항 명세서에서 요구사항을 직접 추출하여 데이터셋을 구성 및 생성하고 이를 통해 요구사항 데이터셋을 추가 및 확장한다. 두번째, 확장된 요구사항 데이터 이해와 분석을 통해 데이터셋을 비교하고 차이점을 분석한다. 세번째, 다양한 자연어처리를 이용하여 새로운 조합을 구성하고 이를 데이터 전처리과정에서 적용 및 변화를 관찰한다. 네번째, 자연어처리를 이용한 데이터 전처리과정으로 생성된 산출물을 기계학습 모델을 적용한다. 이를 통해,

모델의 분류 방법에 대해 먼저 평가하고 그 다음 서로 다른 모델과 성능을 비교하고 분석한다. 마지막으로, 자연어 처리를 이용한 데이터 전처리와 기계학습 모델 사이의 관계를 분석하고 요구사항 분류의 최적의 모델의 도출한다.

### 3.2 Data Understanding and Analyzing

본 연구에서 사용하는 요구사항 데이터는 총 세 가지로 The Quality Attributes (NFR) 데이터세트는 25th IEEE International Requirements Engineering Conference (RE'17)에서 제공하였다. Public Requirements (PURE) 와 Concordia RE Corpus (Concordia) 데이터세트는 본 연구에 검증을 위해 새롭게 생성한 데이터세트로 각각 National Research Council of Italy와 캐나다 Concordia 대학의 The Semantic Software Lab에서 제공 및 연구되었다.

#### 3.2.1 The Quality Attributes (NFR)

The Quality Attributes (NFR) 데이터세트는 실제 DePaul University에서 요구공학을 위해 15개의 팀 소프트웨어 프로젝트를 통해 개발과 작성된 소프트웨어 요구사항 명세서에서 요구사항을 추출한 데이터세트이다. 이 데이터세트는 OpenScience tera-Promise repository와 25th IEEE International Requirements Engineering Conference (RE'17)의 Data track에서 제공 및 소개된 공개 요구사항 데이터세트로, 15개의 소프트웨어의 255개의 기능적 요구사항과 370개의 비기능적 요구사항으로 총 625개의 요구사항이 있다.

NFR 데이터세트는 프로젝트 번호, 자연어로 작성된 요구사항, 그리고 요구사항의 항목, 총 세 개의 데이터 속성 (Attribute)으로 구성되어 있다. 먼저, 프로젝트 번호는 수치형 값(Numeric value)으로 1-15의 값을 가지고 있다. 두 번째, 요구사항은 문자열 값(String value)으로 소프트웨어의 기능적 요구사항과 비기능적 요구사항을 명세하고 있다. 마지막으로, 요구사항의 항목은 본 연구에서 기계학습을 위한 분류 항목으로 기능적 요구사항과 비기능적 요구사항을 포함하여 12개로 구성되어 있다. 각 세부항목이 가진 요구사항의 데이터의 개수는 다음 아래 표 1와 같이 정리된다.

Table 1. NFR Dataset Requirements

| Requirement                |                   |                   | Total |
|----------------------------|-------------------|-------------------|-------|
| Functional Requirement     |                   |                   | 255   |
| Non-functional Requirement | Quality Attribute | Availability      | 21    |
|                            |                   | Look & Feel       | 38    |
|                            |                   | Maintainability   | 17    |
|                            |                   | Operability       | 62    |
|                            |                   | Performance       | 54    |
|                            |                   | Scalability       | 21    |
|                            |                   | Security          | 66    |
|                            |                   | Usability         | 67    |
|                            |                   | Fault-Tolerance   | 10    |
|                            |                   | Portability       | 1     |
|                            | Constraint        | Legal & Licensing | 13    |
| Total                      |                   |                   | 625   |

먼저, 전체 요구사항은 기능적 요구사항과 비기능적 요구사항으로 분류된다. 그리고 비기능적 요구사항은 품질 속성과 제약사항으로 다시 구분된다. NFR 데이터세트의 품질 속성은 가용성, 직관성, 보전성, 조작성, 성능, 확장성, 보안성, 유용성, 내고장성, 그리고 이식성으로, 제약사항은 법률 및 허용에 관한 요구사항이 있다.

#### 3.2.2 Concordia RE Corpus (Concordia)

Concordia RE Corpus 데이터세트는 캐나다 Concordia 대학의 The Semantic Software Lab에서 4개의 소프트웨어 요구사항 명세서, Supplementary Specs 그리고 USECASE를 제공하였고 제공된 문서에서 요구사항을 추출하여 본 연구의 검증을 위해 새롭게 생성한 데이터세트이다. Rashwan et al.[13] 에서 요구사항 분류에 관한 연구를 위해 NFR와 같이 소개 및 사용되었고 이는 778개의 기능적 요구사항과 241개의 비기능적 요구사항으로 총 1019개의 요구사항이 있다.

Concordia 데이터세트는 자연어로 작성된 요구사항, 그리고 요구사항의 항목, 총 두개의 데이터 속성으로 구성되어 있다. 먼저, 요구사항은 문자열 값으로 소프트웨어의 기능적 요구사항과 비기능적 요구사항을 명세하고 있다. 또한, 요구사항의 항목은 본 연구에서 기계학습을 위한 분류 항목으로 기능적 요구사항과 비기능적 요구사항을 포함하여 11개로 구성되어 있다. 각 세부항목이 가진 요구사항의 데이터의 개수는 다음 아래 표 2와 같이 정리된다.

Table 2. Concordia Dataset Requirements

| Requirement                |                   | Total             |     |
|----------------------------|-------------------|-------------------|-----|
| Functional Requirement     |                   | 778               |     |
| Non-functional Requirement | Quality Attribute | Functionality     | 16  |
|                            |                   | Usability/Utility | 28  |
|                            |                   | Efficiency        | 14  |
|                            |                   | Security          | 58  |
|                            |                   | Reliability       | 9   |
|                            |                   | Portability       | 1   |
|                            |                   | Maintainability   | 3   |
|                            |                   | Dependability     | 1   |
|                            |                   | Quality in use    | 1   |
|                            | Constraint        |                   | 110 |
| Total                      |                   | 1019              |     |

먼저, 전체 요구사항은 기능적 요구사항과 비기능적 요구사항으로 분류된다. 그리고 비기능적 요구사항은 품질 속성과 제약사항으로 다시 구분된다. Concordia 데이터 세트의 품질 속성은 기능성, 사용성 및 유용성, 효율성, 보안성, 신뢰성, 이식성, 유지보수성, 신인성, 그리고 사용 품질로, 제약사항과 함께 세부항목으로 구분 되어있다.

### 3.2.3 Public Requirements (PURE)

Public Requirements (PURE) 데이터세트는 공개된 79개의 소프트웨어 요구사항 명세서에서 요구사항을 추출하여 본 연구의 검증을 위해 새롭게 생성한 데이터세트이다. Ferrari et al. [14] 에서 79개의 소프트웨어 요구사항 명세서에 대해 요구사항을 포함하여 명세서가 가진 모든 문장에 대해 분석되었다. 실제 79개 명세서에서 1234개의 기능적 요구사항과 805개의 비기능적 요구사항으로 총 2039개의 요구사항이 추출될 수 있었고 이를 통해, 본 연구에서 새롭게 데이터세트를 생성 및 구성하고 있다. 각 세부항목이 가진 요구사항의 데이터의 개수는 다음 아래 표 3와 같이 정리된다.

Table 3. PURE Requirements dataset

| Requirement                |                   | Total           |     |
|----------------------------|-------------------|-----------------|-----|
| Functional Requirement     |                   | 1234            |     |
| Non-functional Requirement | Module            | 29              |     |
|                            | Quality Attribute | Usability       | 225 |
|                            |                   | Availability    | 21  |
|                            |                   | Performance     | 108 |
|                            |                   | Scalability     | 5   |
|                            |                   | Interface       | 68  |
|                            |                   | Safety          | 10  |
|                            |                   | Security        | 78  |
|                            |                   | Efficiency      | 5   |
|                            |                   | Maintainability | 22  |
|                            |                   | Portability     | 5   |
|                            |                   | Testability     | 4   |
|                            |                   | Reliability     | 40  |
|                            |                   | Supportability  | 8   |
|                            |                   | Serviceability  | 4   |
|                            |                   | Installation    | 8   |
|                            |                   | Operation       | 39  |
|                            |                   | Quality         | 8   |
|                            | Organization      | 104             |     |
|                            | Accessibility     | 2               |     |
| Look & Feel                | 14                |                 |     |
| Capacity                   | 3                 |                 |     |
| Adaptation                 | 2                 |                 |     |
| Constraint                 |                   | 93              |     |
| Total                      |                   | 2039            |     |

먼저, 전체 요구사항은 기능적 요구사항과 비기능적 요구사항으로 분류된다. 그리고 비기능적 요구사항은 모듈, 품질 속성 그리고 제약사항으로 구분된다. 모듈성은 도움 모듈, 지원 모듈, 검사 모듈, 그리고 접근 모듈로, 품질 속성은 이용편의성, 사용성, 가용성, 성능, 확장성, 인터페이스, 안정성, 보안성, 효율성, 유지보수성, 이식성, 시험가능성, 신뢰성, 지원성, 서비스 가용성, 설치, 디자인, 조작성, 보고, 품질, 구조성, 접근성, 직관성, 수용성, 그리고 적응성으로, 제약사항은 법률, 허용 그리고 제약사항으로 세부적으로 구분된다.

### 3.3 Data Preprocessing

본 연구에서 데이터 이해와 분석을 기반으로 데이터 정제 다음으로 자연어처리 기반의 데이터 전처리 수행한다. 데이터 전처리는 두 단계로 진행된다. 먼저, 첫 번째, 오타 및 문자 코드 오류 수정하는 과정이다. 요구사항 데이터는 텍스트 형식으로 구성되어 있고 이에 따라 오타와 문자 코드 오류가 발견되었다. 이는 자연어처리 기법으로 해결되지 않아 때문에 지도학습 과정에서 방해 요소가 된다. 두 번째, 다양한 자연어처리 기법을 이용하여 조합하고 요구사항 데이터에 적용한다. 이를 통해, 텍스트 형식으로 구성된 요구사항 데이터를 지도학습 분류 모델에 적합한 형태로 변환한다.

4. NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements specify the qualitative attributes such as user-friendliness and performance of the system that are critical for the increased user-acceptance of the application.

| Help Module    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.             | The solution should provide detailed context-sensitive help material for all the possible actions and scenarios on all user interfaces in the application.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 2.             | The help should be accessible to the users both in the offline and online mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Support Module |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 1.             | The solution should provide an interface for the user to log any defects or enhancement requests on the application and track thereafter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2.             | The solution should send alerts (e.g., email, SMS) to the user if the user chooses to whenever any action has been taken on the alert                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 3.             | The solution should enable the user to track the submitted defect or enhancement request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 4.             | The solution should enable the help-desk user to view the reports on the submitted defects or enhancement requests category-wise, status-wise, and age-wise.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 5.             | The support solution should be accessible to the users both from within the application and also outside the application through a browser interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Audit Module   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 1.             | <p>An audit trail is a record of actions taken by either the user or the system triggers. This includes actions taken by users or Administrators, or actions initiated automatically by the system as a result of system parameters. The System must keep an unalterable audit trail capable of automatically capturing and storing information about:</p> <ul style="list-style-type: none"> <li>All the actions (create/read/update/delete) that are taken upon the critical entities (case, suspect, property,...) in the system</li> <li>The user initiating and or carrying out the action;</li> <li>The date and time of the event.</li> </ul> |

Fig. 2. Example of requirements in SRS

3.3.1 Natural Language Processing (NLP)

기계학습 모델은 자연어를 입력 값으로 가질 수 없기 때문에 기계학습 모델에 적용하기 위해서는 자연어처리를 통해 단어 표현(Word Representation)으로 변환은 필수적이다. 본 연구의 데이터 전처리 과정에서는 어간 추출(Stemming), 불용어제거(Stopword Removal) 그리고 단어 빈도수-역문서 빈도수(Term Frequency-Inverse Document Frequency, tf-idf)를 사용하고 이를 통해 나온 결과를 바탕으로 정규 표현식을 이용한 토큰화와 토큰을 수치로 표현하는 단어 표현은 고정으로 설정하고 단어 빈도수-역문서 빈도수, Snowball 어간 추출[15], 그리고 불용어제거를 조합한다. 이를 통해, 다음 아래 표 4 과 같이 조합을 총 7개가 생성 및 구성한다.

Table 4. Combination of preprocessing

| Num | Preprocessing                        |
|-----|--------------------------------------|
| 1   | Stemming                             |
| 2   | Stopword removal                     |
| 3   | Stopword removal + Stemming          |
| 4   | tf-idf                               |
| 5   | tf-idf + Stemming                    |
| 6   | tf-idf + Stopword removal            |
| 7   | tf-idf + Stopword removal + Stemming |

3.3.2 Result of Preprocessing

본 연구에서 자연어처리를 통해 생성된 7개의 조합을 NFR, Concordia 그리고 PURE 데이터셋에 적용하여 도출된 토큰과 그 종류의 결과를 다음 표 5로 정리한다.

Table 5. Results of preprocessing

| Dataset   | Preprocessing               | Token  |       |
|-----------|-----------------------------|--------|-------|
|           |                             | Total  | Type  |
| NFR       | Original                    | 12,451 | 1,664 |
|           | Stemming                    | 12,451 | 1,219 |
|           | Stopword removal            | 7,692  | 1,568 |
|           | Stopword removal + Stemming | 7,789  | 1,131 |
| Concordia | Original                    | 11,838 | 1,188 |
|           | Stemming                    | 11,838 | 887   |
|           | Stopword removal            | 7,350  | 1,093 |
|           | Stopword removal + Stemming | 7,488  | 802   |
| PURE      | Original                    | 41,356 | 3,592 |
|           | Stemming                    | 41,356 | 2,329 |
|           | Stopword removal            | 25,898 | 3,477 |
|           | Stopword removal + Stemming | 26,161 | 2,229 |

또한, 요구사항 “Only adjusters can request recycled parts audit reports. No users without an adjuster role shall request recycled parts audits.”으로 이용하여 본 연구에서 7개의 자연어처리 조합으로 데이터 전처리 과정 후의 데이터 형태 변화도 다음 표 6와 같이 정리한다.

3.4 Supervised Learning

본 연구에서 사용되는 데이터셋 내의 모든 데이터는 명시적인 정답 즉, 지도학습에 필요한 항목이 모두 존재한다. 이는 지도학습이 가능하다는 것을 의미하고 본 연구에서는 지도학습을 이용하여 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류한다. 본 연구 실험에서 사용되는 분류 모델로 확률적 모델, 비확률적 모델 그리고 신경망 모델이 있다. 각 분류 모델은 매개변수에 따라 분류 성능과 학습에 영향을 미친다. 이를 고려하여 각 모델에 분류 방식을 다양하게 설정하여 연구 실험에 적용하였다.

Table 6. Comparison between preprocessed data

| Preprocessing | Preprocessed data                                                                                                                                                                                                     |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Original      | Only adjusters can request recycled parts audit reports. No users without an adjuster role shall request recycled parts audits.                                                                                       |
| 1             | shall, user, can, onli, report, no, an, without, request, part, adjust, audit, recycl, role                                                                                                                           |
| 2             | shall, users, reports, without, request, audit, parts, recycled, adjusters, adjuster, audits, role                                                                                                                    |
| 3             | shall, user, onli, report, without, request, part, adjust, audit, recycl, role                                                                                                                                        |
| 4             | shall=0.279, can=3.598, only=2.768, users=2.242, no=3.064, an=3.030, reports=4.485, without=4.033, request=7.585, audit=3.792, parts=6.592, recycled=6.774, adjusters=4.822, adjuster=4.822, audits=6.431, role=4.822 |
| 5             | shall=0.279, user=1.483, can=3.598, onli=2.768, report=2.935, no=3.064, an=3.030, without=4.033, request=6.681, part=6.060, adjust=7.196, audit=7.317, recycl=6.774, role=4.822                                       |
| 6             | shall=0.279, users=2.242, reports=4.485, without=4.033, request=7.585, audit=3.792, parts=6.592, recycled=6.774, adjusters=4.822, adjuster=4.822, audits=6.431, role=4.822                                            |
| 7             | shall=0.279, user=1.483, onli=2.768, report=2.935, without=4.033, request=6.681, part=6.060, adjust=7.196, audit=7.317, recycl=6.774, role=4.822                                                                      |

### 3.4.1 Bayesian Classification Model

베이지 분류 모델은 확률적 분류 모델로 특성들 사이의 독립을 가정하는 베이지 이론을 기초하여 각 요구사항에 사용되는 단어의 출현 빈도에 대해 독립성을 가정하고 각 단어가 레이블에 사용되는 확률을 바탕으로 요구사항을 분류한다. 본 연구에서 사용하는 베이지 분류 모델은 가우시안 분포의 나이브와 다항 분포 베이지를 적용한다.[16]

### 3.4.2 Support Vector Machine (SVM)

서포트 벡터 머신은 비확률적 분류 모델로, 커널 트릭을 이용하여 서포트 벡터 머신 분류 모델이 요구사항 데이터를 비확률적으로 선형과 비선형 분류를 가능하다. 이를 통해, 주어진 요구사항이 기능적 요구사항 또는 비기능적 요구사항 항목에 범주화하는지 판단하기 위해 선형과 비선형으로 분류할 수 있도록 커널 트릭을 이용하여 요구사항을 비확률적으로 선형 분류와 비선형에서 시그모이드, 방산형 그리고 다항 커널을 설정하여 비선형 분류, 총 4가지 서포트 벡터 머신을 본 연구에서 사용한다.

### 3.4.3 Artificial Neural Network (ANN)

신경망 모델은 입력 계층은 자연어처리 기법 조합을 이용한 전처리가 끝난 후 토큰의 개수로, 은닉 계층은 하나와 노드 수 100으로, 그리고 출력 계층은 분류할 항목 개수로 설정하고 최적화 알고리즘으로 Adam으로 설정하고, 활성화 함수를 선형, 시그모이드, 탄젠트형, 그리고 리루를 설정하여 총 4가지 신경망 모델을 구성하고 이를 요구사항 분류에 적용한다.

### 3.5 Analyzing and Comparing

본 연구에서 진행한 연구 실험의 결과는 정밀도와 재현율을 기준으로 자연어처리와 기계학습 분류 모델을 비교한다. 먼저, 모든 결과는 k겹 교차 검증 방식(k-Fold Cross Validation)을 통하여 출력된다. k겹 교차 검증 방식은 전체 데이터 세트를 k개의 그룹으로 분할하고(k-1)개의 학습 그룹과 1개의 검증 그룹으로 지정하고 검증마다 학습 그룹과 검증 그룹을 다르게 하여 총 k회 검증을 통해 성능을 측정한다. 이를 통해, 정밀도는 참으로 예측된 요구사항 중 실제로 참인 요구사항의 비율을 말한다. 재현율은 실제 참인 요구사항들 중 정확히 참인 요구사항으로 식별된 비율을 말한다.

#### 3.5.1 Result and Analysis of Bayesian Model

Table 7. Bayesian classification results

| Dataset   | Model       | NLP   | Precision | Recall |
|-----------|-------------|-------|-----------|--------|
| NFR       | Naive       | 1     | 0.839     | 0.837  |
|           | Multinomial | 1     | 0.897     | 0.898  |
| Concordia | Naive       | 3     | 0.888     | 0.859  |
|           | Multinomial | 4     | 0.878     | 0.876  |
| PURE      | Naive       | 1 / 5 | 0.779     | 0.776  |
|           | Multinomial | 4     | 0.873     | 0.872  |

먼저, 표 7에서 베이지 분류 모델에서 사용된 나이브 베이지와 다항 분포 베이지를 비교 분석한다. 전체적으로 결과를 비교하면 다항 분포 베이지가 나이브 베이지보다 좋은 결과를 도출할 수 있었다. 이는, 데이터의 특성을 계산할 때, 가우시안 분포로 계산하는 나이브 베이지보다 다항

분포로 계산하는 다항 분포 베이스 모델이 같은 베이스 이론에 기반한 모델이라도 확률에 대해 특성의 분포에 대해 그 분류 성능은 2% ~ 12% 차이를 보여주고 있다. 이를 통해, 요구사항 분류와 더 나아가 문서 분류에 있어서 모든 데이터 전처리 경우에서 다항 분포 베이스에서 더 좋은 결과를 도출할 수 있다.

### 3.5.2 Result and Analysis of SVM

Table 8. SVM classification results

| Dataset   | Kernel     | NLP | Precision | Recall |
|-----------|------------|-----|-----------|--------|
| NFR       | Linear     | 4   | 0.852     | 0.848  |
|           | Sigmoid    | 2   | 0.773     | 0.712  |
|           | RBF        | 2   | 0.774     | 0.702  |
|           | Polynomial | 3   | 0.786     | 0.725  |
| Concordia | Linear     | 1   | 0.831     | 0.831  |
|           | Sigmoid    | 1   | 0.799     | 0.575  |
|           | RBF        | 4   | 0.733     | 0.624  |
|           | Polynomial | 1   | 0.671     | 0.297  |
| PURE      | Linear     | 3   | 0.708     | 0.697  |
|           | Sigmoid    | 2   | 0.695     | 0.51   |
|           | RBF        | 2   | 0.678     | 0.487  |
|           | Polynomial | 2   | 0.727     | 0.531  |

두번째, 서포트 벡터 머신을 이용한 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류하는 연구 실험에서 요구사항 데이터를 비확률적인 선형과 비선형으로 분류하기 위해 커널을 선형과 시그모이드, 방산형 그리고 다항으로 3개의 비선형 커널을 설정하고 비교 분석한다. 표 8은 각각 자연어처리와 기계학습 모델을 통해 가장 좋은 결과의 선형 분류와 비선형 분류의 결과를 정밀도와 재현율로 정리한다. 전체적으로 서포트 벡터 머신으로 비확률적 선형 이진 분류의 결과가 비선형 이진 분류보다 큰 차이로 높은 성능의 결과가 도출되었다. 이는, 요구사항과 같은 문자열 데이터 또는 문서 분류에서는 자연어처리를 이용한 다양한 데이터 전처리를 진행해도 선형 분류는 비선형의 분류 결과보다 적용하는 것이 좋고 이는 서포트 벡터 머신에서 비선형 커널은 본 연구 문제를 해결하는데 있어서 적합한 모델이 아닌 것을 보여주고 있다. 또한, 비선형 분류 과정에서도 같은 요구사항 데이터라도 데이터가 가진 특징과 성격에 따라서 그 분류 결과는 다양하게 도출된 것을 확인할 수 있었다.

### 3.5.3 Result and Analysis of ANN

Table 9. ANN classification results

| Dataset   | Activation | NLP | Precision | Recall |
|-----------|------------|-----|-----------|--------|
| NFR       | tanh       | 4   | 0.909     | 0.909  |
|           | Sigmoid    | 4   | 0.907     | 0.907  |
|           | Linear     | 4   | 0.905     | 0.904  |
|           | ReLu       | 4   | 0.909     | 0.909  |
| Concordia | tanh       | 5   | 0.856     | 0.856  |
|           | Sigmoid    | 5   | 0.852     | 0.854  |
|           | Linear     | 5   | 0.852     | 0.853  |
|           | ReLu       | 5   | 0.857     | 0.857  |
| PURE      | tanh       | 4   | 0.892     | 0.892  |
|           | Sigmoid    | 4   | 0.895     | 0.896  |
|           | Linear     | 6   | 0.89      | 0.89   |
|           | ReLu       | 6   | 0.892     | 0.892  |

세번째, 신경망을 이용한 요구사항 분류에서 신경망의 활성화함수를 탄젠트, 시그모이드, 선형, 그리고 리루 함수로 설정하고 비교 분석한다. 표 9은 다양한 활성화함수를 이용한 신경망 모델의 가장 좋은 분류 결과를 정밀도 그리고 재현율로 정리한다. 본 연구의 실험 결과로, 모든 요구사항 데이터셋에 대해 단어 빈도수-역문서 빈도수는 기계학습 분류 성능에 가장 필수적인 요인이지만 오히려 어간 추출과 불용어 제거와 같은 단어의 개수와 종류에 영향을 주는 자연어처리를 같이 쓰는 경우는 신경망 모델의 학습에 방해가 되는 것을 확인할 수 있었다.

### 3.5.4 Result and Analysis of Models

Table 10. Total results

| Dataset   | Model       |             | NLP | Precision | Recall |
|-----------|-------------|-------------|-----|-----------|--------|
| NFR       | Multinomial |             | 1   | 0.897     | 0.898  |
|           | SVM         | Linear      | 4   | 0.852     | 0.848  |
|           | ANN         | tanh / ReLu | 4   | 0.909     | 0.909  |
| Concordia | Multinomial |             | 4   | 0.878     | 0.876  |
|           | SVM         | Linear      | 1   | 0.831     | 0.831  |
|           | ANN         | ReLu        | 5   | 0.857     | 0.857  |
| PURE      | Multinomial |             | 4   | 0.873     | 0.872  |
|           | SVM         | Linear      | 3   | 0.708     | 0.697  |
|           | ANN         | Sigmoid     | 4   | 0.895     | 0.896  |



표 10는 본 연구에서 사용된 모든 데이터셋에 대해 자연어처리와 기계학습을 이용한 요구사항분류에 대해 가장 좋은 결과를 정리하였다. NFR 데이터셋에서는 단어 빈도수-역문서 빈도수로 데이터 전처리를 적용 후에 탄젠트와 리루 활성화함수를 이용한 신경망 모델이, Concordia 데이터셋에서는 단어 빈도수-역문서 빈도수로 데이터 전처리를 적용 후에 다항 분포 베이스 분류 모델이 그리고 PURE 데이터셋에서는 단어 빈도수-역문서 빈도수로 데이터 전처리를 적용 후에 시그모이드 활성화함수를 이용한 신경망이 가장 성능을 도출할 수 있었다. 이를 통해, 신경망 모델은 요구사항의 개수와 요구사항의 문장 구성이 복잡해진 경우에 베이스 분류 모델과 서포트 벡터 머신보다 정확한 분류가 유추 가능하고 요구사항의 개수는 많지만 문장의 구성이 단순한 경우에는 신경망 모델보다 베이스 분류 모델을 통해 정확한 분류가 가능하다는 것을 유추할 수 있다. 자연어처리를 이용한 데이터 전처리에서는 어간 추출과 불용어제거와 같은 토큰의 개수와 종류를 감소하여 데이터의 희소성을 좀 더 밀집형태로 변형하는 데이터 전처리보다는 단어 빈도수와 역문서 빈도수를 기반으로 단어의 가중치를 계산하는 데이터 전처리가 다른 전처리보다 좋은 결과를 도출할 수 있었다. 이를 통해, 모든 단어를 고려하여 가중치 값은 기계학습에서 긍정적인 요인을 볼 수 있고 오히려 문장에서 의미 없는 단어를 제거하는 불용어 제거는 부정적인 요소로 확인할 수 있었다.

## IV. Conclusion and Future work

### 4.1 Conclusions

본 연구는 소프트웨어 요구사항 명세서에서 추출된 요구사항을 인공지능의 자연어처리와 기계 학습을 이용하여 데이터 중심적 접근법을 기반으로 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류에 대해 연구를 진행하였다. 본 연구에서는 관련 연구와의 다른 자연어처리와 기계학습 모델을 통해 연구 비교 분석 연구하였다. 이를 통해, 요구사항 분류에 대하여, 관련 연구와는 다른 방법을 제시하고 데이터 전처리 과정에서 자연어처리 조합과 지도 학습의 분류 모델을 적용하고 그 결과를 도출하였다. 마지막으로, 새로운 요구사항 데이터를 분류 모델을 통해 연구 실험 결과에 대해 검증 및 확인을 하였다.

### 4.2 Contributions

본 연구의 기여도는 크게 세 가지가 있다. 첫째, 새로운 데이터셋 생성을 통한 데이터 확장이다. 이는, 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류하는 연구에서 사용되는 데이터의 규모는 너무 작아서, 기계학습을 이용한 데이터 중심적 접근법을 통한 요구사항 분류의 연구에서 제한이 많이 있었다. 즉, 본 연구에서는 수 많은 소프트웨어 요구사항 명세서에서 요구사항을 추출, 수집 그리고 데이터셋으로 구성 및 생성하여 새로운 요구사항 데이터셋을 연구에 적극 활용하여 기존의 데이터 규모에 대한 문제점과 연구의 한계점을 해결하였다.

두 번째, 다양한 요구사항 데이터에 대해 자연어처리와 기계학습 모델의 비교 분석이다. 본 연구에서 관련 연구와는 달리 새로운 생성한 요구사항 데이터셋을 연구에 활용하였고, 이를 통해, 관련 연구와는 다른 수의 데이터와 양질의 요구사항 데이터를 연구 실험에서 적극 활용할 수 있었다. 그 결과, 본 연구에서는 데이터의 수와 질이 어느정도 보장 되어있는 경우에 자연어처리와 기계학습 모델 적용이 어떤 결과를 도출할 수 있는지 실험을 통해 확인할 수 있었고, 본 연구를 통해 데이터의 수와 질이 기계학습 적용 결과에 어떤 영향을 미치는지 비교 분석을 할 수 있었다.

세 번째, 다양한 자연어처리와 기계학습 모델을 적용하여 요구사항 분류에 대해 결과와 최적화된 모델을 제시하였다. 본 연구에서 요구사항 데이터에 대해 자연어처리를 이용한 다양한 데이터 전처리 조합과 기계학습 모델을 적용하여 그 결과를 도출하였고 이를 비교 분석을 통해 본 연구의 문제점 해결에 최적화된 방법을 제시하고 있다. 이를 통해, 향후 요구사항 데이터를 활용한 연구에서 본 연구에서 도출된 결과와 제시하는 최적화된 방법을 활용할 수 있다.

### 4.3 Limitations

본 연구의 한계점은 크게 두 가지로 구분된다. 첫째, 본 연구 실험의 결과에 있다. 기계 학습을 이용한 연구에서 정확도, 정밀도 그리고 재현율과 같이 모델의 성능을 평가하는 기준의 점수가 높을수록 그 모델의 성능은 좋다고 평가할 수 있다. 하지만 본 연구에서 다양한 데이터 전처리와 기계학습 모델을 사용하여 다양한 결과를 도출하였지만 관련 연구의 결과보다 뚜렷하게 향상된 결과를 보여주지 못했다.

두 번째, 본 연구 접근법에 있다. 본 연구에서 자연어처리와 기계학습 모델, 그리고 각 모델의 다양한 설정을 통해 모델의 분류 결과에 대해 비교 분석하였다. 본 연구에서 자연어처리와 기계학습을 이용한 데이터 중심적 접근법으로 연

구 문제에 대해 해결책을 제시하고 있지만, 그러나 추후 다양한 방법으로 문제를 해결할 수 있고 본 연구에서 서로 다른 접근법과의 비교 분석 및 평가를 보여주지 못했다.

#### 4.4 Future Works

향후 연구에서는 크게 세 가지를 고려할 수 있다. 첫 번째, 본 연구의 한계점을 극복하는 방향으로 연구를 진행할 수 있다. 본 연구에서는 소프트웨어 요구사항 명세서에서 요구사항 추출하여 새로운 요구사항 데이터셋을 생성과 동시에 데이터 중심적인 접근법을 통해 자연어처리와 기계학습을 적용하였기 때문에 연구의 접근법에 한계가 있다. 이는, 다른 연구 접근법 또는 다른 분석을 통해 요구사항 분류에 대한 연구를 다양한 측면에서 진행하고 이를 비교 및 검증한다면 더욱 의미 있는 결과를 도출할 수 있기 때문이다.

두 번째, 본 연구에서 다루지 못한 요구사항의 문제를 해결하는 방향을 진행할 수 있다. 본 연구에서는 요구사항을 기능적 요구사항과 비기능적 요구사항으로 분류에만 집중하였기 때문에 요구사항이 가지는 다양한 문제점을 다루지 못하였다. 이는, 다양한 요구사항들 간에 존재하는 문제에 대해 연구와 분석이 필요하며 이를 통해, 본 연구에서 제안하는 방법보다 더 정확하고 적합한 연구 문제 해결 방법을 제시할 수 있다.

마지막으로, 본 연구의 실험 방법을 다른 도메인에 적용하는 방향을 고려할 수 있다. 본 연구에서는 특정 항목에 대해 다양한 자연어처리와 기계학습 모델을 통해 데이터 전처리와 분류를 연구 실험에서 진행하였고 이를 통해, 실험 과정에서 다양한 데이터의 변화와 실험 결과를 도출할 수 있었다. 이와 같이, 특정 항목에 대해 분류하는 연구에 대해 본 연구에서 정형화된 연구 방법론을 적용할 수 있으며 이는, 다양한 도메인의 문제를 해결할 수 있는 방법으로 제시할 수 있다.

#### ACKNOWLEDGEMENT

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2020R1F1A1075605).

#### REFERENCES

- [1] Bourque, Pierre, and Richard E. Fairley. Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press, 2014.
- [2] Kotonya, Gerald, and Ian Sommerville. Requirements engineering: processes and techniques. Wiley Publishing, 1998.
- [3] Sommerville, Ian. "Software engineering 9th Edition." ISBN-10 137035152 (2011).
- [4] Glinz, Martin. "A glossary of requirements engineering terminology." Standard Glossary of the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version 1 (2011).
- [5] Van Lamsweerde, Axel. Requirements engineering: From system goals to UML models to software. Vol. 10. Chichester, UK: John Wiley & Sons, 2009.
- [6] Ernst, Neil A., and John Mylopoulos. "On the perception of software quality requirements during the project lifecycle." International Working Conference on Requirements Engineering: Foundation for Software Quality. Springer, Berlin, Heidelberg, 2010.
- [7] Abad, Zahra Shakeri Hossein, and Guenther Ruhe. "Using real options to manage technical debt in requirements engineering." 2015 IEEE 23rd International Requirements Engineering Conference (RE). IEEE, 2015.
- [8] Abad, Zahra Shakeri Hossein, et al. "What are practitioners asking about requirements engineering? an exploratory analysis of social q&a sites." 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW). IEEE, 2016.
- [9] Abad, Zahra Shakeri Hossein, et al. "What works better? a study of classifying requirements." 2017 IEEE 25th International Requirements Engineering Conference (RE). IEEE, 2017.
- [10] Hussain, Ishrar, Leila Kosseim, and Olga Ormandjieva. "Using linguistic knowledge to classify non-functional requirements in SRS documents." International Conference on Application of Natural Language to Information Systems. Springer, Berlin, Heidelberg, 2008.
- [11] Dekhtyar, Alex, and Vivian Fong. "RE data challenge: Requirements identification with Word2Vec and TensorFlow." 2017 IEEE 25th International Requirements Engineering Conference (RE). IEEE, 2017.
- [12] Y. Kim, "Convolutional neural networks for sentence classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 2014, pp. 1746-1751.
- [13] Rashwan, Abderahman, Olga Ormandjieva, and René Witte. "Ontology-based classification of non-functional requirements in software specifications: a new corpus and svm-based classifier."

2013 IEEE 37th Annual Computer Software and Applications Conference. IEEE, 2013.

- [14] Ferrari, Alessio, Giorgio Ortonzo Spagnolo, and Stefania Gnesi. "PURE: A dataset of public requirements documents." 2017 IEEE 25th International Requirements Engineering Conference (RE). IEEE, 2017.
- [15] Porter, Martin F. "Snowball: A language for stemming algorithms." (2001).
- [16] Qiang, Guo. "An effective algorithm for improving the performance of Naive Bayes for text classification." 2010 Second International Conference on Computer Research and Development. 2010.

## Authors



Byung-Sun Cho received the B.S. in Information Technology from Saint Louis University, Philippines, in 2016 and M.S. in Computer Engineering from Ajou University, Korea in 2020. He is interested in

Software Engineering, Requirements Engineering, Machine Learning and Natural Language Processing



Seok-Won Lee received the BS in Computer Science from Dongkuk University at Seoul, Republic of Korea. MS in Computer Science from University of Pittsburgh at Pittsburgh, Pennsylvania, and PhD in Information

Technology from George Mason University at Fairfax, Virginia, U.S.A. He has been a faculty member at the University of North Carolina at Charlotte and the University of Texas at San Antonio. He is currently a Full Professor and Chair of the Department of Software and Compute Engineering at Ajou University, Republic of Korea. His areas of specialization include software engineering with specific expertise in ontological requirement engineering and domain modeling, and knowledge engineering with specific expertise in knowledge acquisition, machine learning and knowledge-based systems.