

NLP 기계 학습을 사용한 한글 요구사항 문서에서의 요구사항 자동 생성 프로세스

백영윤^{*†} · 박수진^{**} · 박용범^{*}

^{*†} 단국대학교 소프트웨어학과, ^{**} 서강대학교 기술경영전문대학원

Process for Automatic Requirement Generation in Korean Requirements Documents using NLP Machine Learning

Young Yun Baek^{*†}, Soo Jin Park^{**} and Young Bum Park^{*}

^{*†} Dept. of Software Science Dankook University,

^{**} Graduate School of Management of Technology, Sogang University

ABSTRACT

In software engineering, requirement analysis is an important task throughout the process and takes up a high proportion. However, factors that fail to analyze requirements include communication failure, different understanding of the meaning of requirements, and failure to perform requirements normally. To solve this problem, we derived actors and behaviors using morpheme analysis and BERT algorithms in the Korean requirement document and constructed them as ontologies. A chatbot system with ontology data is constructed to derive a final system event list through Q&A with users. The chatbot system generates the derived system event list as a requirement diagram and a requirement specification and provides it to the user. Through the above system, diagrams and specifications with a level of coverage complied with Korean requirement documents were created.

Key Words : Korean document, BERT, Chatbot, NLP, Requirement diagram, Requirement specification

1. 서 론

소프트웨어 공학에서 요구 분석은 소프트웨어 개발에 있어서 만족해야 할 요구사항을 도출, 분석, 명세화, 검증하는 과정으로 개발 초기 단계에서 중요한 프로세스이다. 이를 통해서 만들어질 소프트웨어의 기능이나 목표, 제약 사항 등을 사용자로부터 시스템 개발에 필요한 내용을 알아 낼 수 있다. 따라서 요구 분석은 소프트웨어 개발에서 사용자가 원하는 소프트웨어를 만드는 것에 있어서 중요하고 또한 소프트웨어 개발 주기에서 가장 중요한 과정이다[1].

하지만 요구사항 기술서 문서에서 명확한 요구사항 내

용을 추출하고 소프트웨어 전반을 커버하는 요구사항 내용을 추출하는 방법은 여전히 어렵고 복잡하다. 또한 이러한 요구사항을 추출하는 과정에서 사람이 작업을 하므로 요구사항 추출의 일관성과 인적 작업으로 인한 분석 제한이 존재한다.

본 논문에서는 이러한 문제를 해결하기 위해서 문제 기술서 단계의 문서에서 인공지능 분석과 자연어 처리를 통해 핵심 단어를 추출하고 이를 BERT Q&A를 통해서 액터와 행동 간 관계를 확보하여 이를 온톨로지 구성하여 챗봇에 기본 데이터로써 사용한다. 이를 통해 복잡한 요구사항 기술서에서 일관된 요구사항 내용을 도출할 수 있으며, 사용자에게 챗봇을 통한 질의 응답을 통해서 요구사항 다이어그램과 요구사항 명세서를 자동으로 생성하여서 인적 작업을 해결하는 방법을 제안한다.

[†]E-mail: youngyun.baek@dankook.ac.kr

2. 관련 연구

2.1 규칙, 분류 통한 요구사항 분석 방법

[2]에서는 자연어 요구 사항 문서에서 규칙기반 접근 방식을 사용하여 목표 및 Use-Case 모델을 자동으로 추출하는 방식을 연구하여 이해도, 인적 작업 문제를 해결하였다. [3]에서는 요구사항 명세를 분류하기 위해서 여러 분류 알고리즘을 사용하여서 각 알고리즘 별로 효율을 측정하여 요구사항을 추출하였다. 이를 통해 커버리지, 인적 작업 문제를 해결하였다. [4]에서는 복잡한 소프트웨어 엔지니어링 문제를 해결하기 위해 자율 코드 생성을 위한 기계 학습 요구 사항에 대한 방법을 제시하여 요구사항을 추출하여 일관성, 인적 작업 문제를 해결하였다.

2.2 머신 러닝을 통한 요구사항 분석 방법

[5]에서는 지능형 대화형 챗봇을 사용하여 요구사항 추출 및 분류를 자동화하는 방법을 제시하여 이해 관계자와 자연어로 대화하고 상호 작용해서 시스템 요구사항을 도출한 다음 요구사항을 추출하였다. 이를 통해 인적 작업 및 일관성 문제를 해결하였다. [6]에서는 요구 사항 사양 및 사용자 피드백 분석 작업을 자동화하는 도구를 만들어 RE 회의에서 논의된 요구사항과 사용자 리뷰 형식의 텍스트 피드백을 자동으로 요구사항 분류할 수 있도록 제시하여 이해도와 인적 작업 문제를 해결하였다. [7]에서는 텍스트 디자인 데이터를 입력 받아 기계 학습 기반의 언어 처리 모델을 적용하여 텍스트 구절을 재귀적으로 분해하여 기능 요구 사항의 구조화된 계층 구조를 추출하는 방법을 제시하여 인적 작업 문제와 커버리지 문제를 해결하였다.

2.3 기타 요구사항 분석 방법

[8][9][10]에서는 요구 분석을 분석하기 위한 여러 방법들을 소개하고 각 방법에 대하여 효과적인 요구사항 추출 알고리즘, 데이터 상태, 각 알고리즘 조합에 따른 요구사항 추출의 최상의 상태를 연구하여 요구사항을 분석하였다.

3. 본 론

3.1 시스템 구성

시스템의 구성은 이전의 논문에서 작성된 구조에서 몇몇 변경과 추가된 사항이 있다[1]. 먼저 기존 머신 러닝 작업에서 수행하려 하였던 BERT 개체명 인식을 사용하지 않고 Mecab 형태소 분석을 통한 자연어 처리를 수행한다. BERT 개체명 인식이 아닌 Mecab 형태소 분석을 사용

하여서 개체명 인식 속도에서 인식하는 속도의 저하는 있으나, 학습 오류로 인한 잘못된 개체명 인식을 방지할 수 있고, 개체명 외의 요구사항을 위한 문장을 생성하는데 있어 Mecab 형태소 분석이 더 유리함을 가지고 있음을 연구하였다. 그 다음 형태소 분석된 내용과 BERT Q&A 동작을 수행하여 온톨로지를 구성하고 온톨로지를 통한 액터, 행동, 이벤트, 시스템 등등을 추출하게 된다. 추출된 형태소 분석 데이터와 온톨로지 데이터는 챗봇 프로그램에 베이스 데이터로써 존재하게 되며, 사용자는 챗봇과 요구사항 문서 Q&A를 통해서 최종적인 요구사항 명세서 작성을 위한 과정을 거치게 된다. 사용자가 최종적으로 챗봇과의 Q&A를 수행하기 때문에 잘못된 요구사항을 선택할 수 있는 확률이 낮아지고 요구사항에 대한 정확도와 커버리지를 확보할 수 있다. 사용자의 경우에는 기존 수동적으로 모든 문장을 분석하는 것이 아닌 챗봇 Q&A가 질문하는 내용에 집중하여 요구사항을 확인할 수 있으므로 요구사항을 분석하는 시간과 요구사항을 선택함에 있어 분석된 데이터를 확인하는 것으로 정확성을 가질 수 있다. 챗봇 Q&A가 종료되면 챗봇 프로그램은 사용자의 응답에 의한 요구사항 명세서 작성과 UML 다이어그램 생성 작업을 수행하고 사용자에게 산출물을 제공한다. 위 시스템 구성은 다음 Fig.1과 같다.

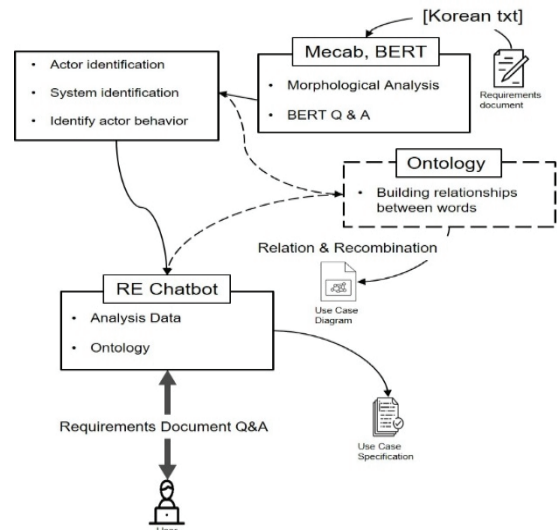


Fig. 1. System Configuration.

3.2 문제 기술서 데이터 및 텍스트 전처리

본 논문에서 사용되는 데이터는 [1]에서 사용하였던 데이터에 추가적인 문제 기술서를 추가하였으며 추가된 문제 기술서를 데이터 또한 자연어 한글로 작성되어 있다.

이 문제 기술서를 텍스트 데이터로 저장하여 관리하며, 텍스트 내에서 발생한 오타 및 띄어 쓰기, 개행에 대한 오류를 수정한다.

3.3 Mecab 형태소 분석

3.3.1 문서 내에서 형태소 분석을 통해 추출 가능한 단어

Mecab 형태소 분석을 통해서 문서 내에서 즉각적으로 뽑아 낼 수 있는 유의미한 단어들은 다음과 같다. 명사 단어, 의미 단어, 형태소 분류 단어, 띄어 쓰기 단어, 주어 단어, 주어 조사, 복수 주어 조사, 목적어 조사, 행동 조사 가 있다. 각 단어들의 추출 방법은 Fig.2와 같다.

- 명사 단어: mecab에서 명사 추출 함수를 사용하여 추출
- 의미 단어: mecab에서 의미 추출 함수를 사용하여 추출
- 형태소 분류 단어: mecab에서 형태소 분석 함수를 사용하여 추출
- 띄어쓰기 단어: 띄어쓰기로 문장을 분리하여 추출
- 주어 단어: 형태소 분석에서 형태소 분류가 "NNG";"NNP";"NNB";"NNBC";"NR";"NP"에 해당하는 단어
- 주어 조사: 형태소 분석에서 형태소 분류가 "JX"에 해당하는 단어
- 복수 주어 조사: 형태소 분석에서 형태소 분류가 "XSN"에 해당하는 단어
- 목적어 조사: 형태소 분석에서 "JKB";"JKO";"JK"에 해당하는 단어

Fig. 2. How to extracted through morpheme analysis within a document.

3.3.2 형태소 분석, 단어 조합을 통하여 추출 가능한 단어

이전 단계에서 추출한 각 단어들과 형태소 분석 분류를 조합하여 뽑아 낼 수 있는 유의미한 단어들은 다음과 같다. 질문사항, 추출 주어, 추출 목적어, 추출 질문 행동, 평균 단어 길이가 있다. 각 단어 및 길이 추출 방법은 Fig.3과 같다.

- 질문: BERT 모델에 입력한 질문사항 문장
- 응답: BERT 모델에서 질문사항에 대해 응답한 문장
- 주어: BERT 모델에 입력한 질문사항 문장의 주어
- 의문사: BERT 모델에 입력한 질문사항 문장의 의문사
- 질문 문장: BERT 모델에 입력한 질문사항 문장의 질문 문장
- 응답 길이: BERT 모델에서 질문사항에 대해 응답한 문장의 길이
- 추출 응답 단어: BERT 모델에서 질문사항에 대해 응답한 문장에서 형태소 분석 시에 조사들을 제외한 나머지 부분만 저장하며, 이는 온톨로지 구성시에 온톨로지에 바로 적용하기 위해 전처리한다.
- 추출 문장: BERT 모델에서 질문사항에 대해 응답한 문장이 나온 원문 부분
- 문장 순서: BERT 모델에서 질문사항에 대해 응답한 문장이 나온 원문 부분의 순서

Fig. 3. How to extracted extracted through word combination.

3.4 BERT Q&A

3.4.1 BERT Q&A 모델 학습

BERT Q&A 기능을 사용하기 위해 다음과 같은 하이퍼 파라미터를 선택하였다. 하이퍼 파라미터는 다양한 실험

을 통하여서 최적의 값을 선정한 것이며, 가장 효과적으로 Q&A를 수행할 수 있는 수치를 작성하였다.

3.4.2 BERT Q&A 모델 수행

학습된 BERT 모델에 이전 형태소 분석을 통해 분석된 문장을 조합하여 질문 사항을 생성한다. 해당 과정은 Fig.4와 같다.

- 질문 사항: 형태소 분석에서 형태소 분류가 "XSV -> EC";"VV+ETM -> NNB";"XSV+ETM -> NNB"의 순서를 가지는 단어 추출
- 추출 주어: 주어 단어 + 주어 조사를 조합하였을 때, 띄어 쓰기 단어와 비교하여 일치하는 단어를 생성한다. 위와 같이 주어를 추출하는 이유는 BERT Q&A의 질문사항 작성시 발생하는 동작 시간을 감소하기 위해 수행한다.
- 추출 목적어: 주어 단어 + 목적어 조사를 조합하였을 때, 띄어 쓰기 단어와 비교하여 일치하는 단어를 생성한다. 위와 같이 주어를 추출하는 이유는 BERT Q&A의 질문사항 작성시 발생하는 동작 시간을 감소하기 위해 수행한다.
- 추출 질문 행동: 질문 사항 단어에서 형태소 분류가 "EC"와 "NNB"에 따라서 분류하여 질문 내용을 생성하고 이를 띄어 쓰기 단어와 비교하여 일치하는 단어를 생성한다. 위와 같이 주어를 추출하는 이유는 BERT Q&A의 질문사항 작성시 발생하는 동작 시간을 감소하기 위해 수행한다.
- 평균 단어 길이: BERT Q&A를 통해서 나오는 응답 문장의 길이를 제한하기 위해 추출하며, 띄어 쓰기 단어 기준으로 평균 단어 길이를 만든다

Fig. 4. How to make BERT Q&A question and answer preprocessing.

3.4.3 BERT Q&A 를 통해 추출되는 결과

BERT Q&A를 통해서 질문, 응답, 주어, 의문사, 질문 문장, 응답 길이, 응답 단어에 대한 추출 응답 단어, 응답 단어가 추출된 문장, 응답 단어가 추출된 문장의 순서를 추출한다. 각 항목에 따른 추출 방법은 Fig.5와 같다.

- 질문 사항: 추출 주어 + 의문사 + 목적어 조사 + 추출 질문 행동
- 생성된 질문 사항은 BERT 모델의 질문 내용으로 선택되며, 문장 전체를 기준으로 질문을 한다. 그 결과 BERT Q&A가 응답으로 찾아낸 결과물을 사용한다. 응답된 결과물의 경우 다음과 같은 전처리를 수행한다.
- 응답 문장에 대하여 [SEP]를 포함하는 경우 [SEP]를 제거한다.
- 응답 문장에 대하여 [CLS]와 [UNK]를 포함하는 경우 해당 응답 문장은 사용하지 않는다.
- 응답 문장이 평균 단어 길이를 넘어가는 경우 해당 응답 문장은 사용하지 않는다.

Fig. 5. How to extracted through BERT Q&A.

3.5 온톨로지 구성

BERT Q&A에서 추출된 주어 단어를 기준으로 온톨로지의 주어부를 작성한다. 이후 추출된 응답 단어와 질문 문장을 조합하여 목적부를 작성한다. 술어부의 경우 고정된 주어에 따른 행동을 작성하는 것이므로 변경 없이 사용한다. 구성된 온톨로지는 RDF, OWL로 사용가능하며 파일을 ttl 확장자로 저장하여 사용한다.

3.6 RE Chatbot 응답을 통한 요구사항 문서 작성

챗봇은 미리 구성된 온톨로지 파일을 읽어와서 액터 정보와 행동 정보를 구성하여 리스트로 제공한다. 사용자는 챗봇이 제공하는 액터 정보와 행동 정보를 보고 실제

요구사항에 해당하는 액터 정보와 행동 정보를 선택하게 된다. 챗봇은 사용자가 선택한 액터와 행동 정보를 기준으로 하여 액터와 행동을 조합하여 리스트를 만들고 다시 사용자에게 제공한다. 사용자는 챗봇이 제공한 리스트를 보고 해당하는 요구사항을 선택하게 된다. 최종적으로 선택된 요구사항에 대해 챗봇은 simple version의 요구사항 명세서 문서를 작성하여 사용자에게 제공한다. 이를 알고리즘으로 만들면 Fig.6과 같다.

4. 요구사항 문서에 적용한 사례 연구

제한된 시스템을 검증하기 위해서 여러 요구사항 문서 중 수강 신청에 관한 요구사항을 선택하여서 논문에서 제안하는 요구사항 다이어그램과 요구사항 명세서 자동 생성 시스템에 적용하여 시스템에 대한 검증을 진행하였다.

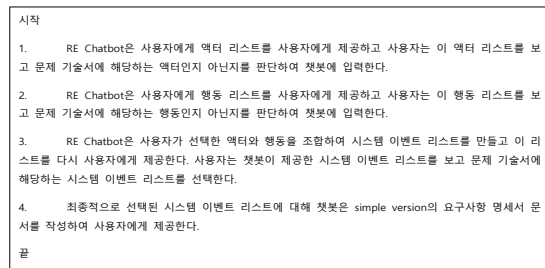


Fig. 6. RE chatbot algorithm.

4.1 문제 기술서

시스템이 적용하기 위해 선택한 문제 기술서의 내용은 다음과 같다. 참고를 위해 영어로 작성하였으나 실제로 한글로 이루어진 문제 기술서를 사용하였다.

4.2 문제 기술서 형태소 분석

문제 기술서에서 각 문장을 mecab 형태소 분석을 수행해 각 단어 별 형태소 태그 결과를 획득한다. 예를 들어 “교수는 시스템에 액세스하여 가르칠 코스를 등록하고 성적을 기록할 수 있다.”라는 문장이라면 “교수”, “시스템”, “액세스”, “코스”, “등록”, “성적”, “기록”이라는 단어들은 “NNG”라는 형태소를 가지고 “는”이라는 단어는 “JX”라는 형태소를 가지고 “에”라는 단어는 “JKO”라는 형태소를 가진다. 이렇듯 각 단어들은 형태소 분석결과 별로 형태소 태그를 가지게 된다.

4.3 형태소 분석에 따른 자연어 처리 수행

형태소 분석된 결과를 통해서 BERT Q&A에서 질의로 사용할 문장을 만들기 위해 주어, 목적어, 질문을 추출한

다. 해당 단어들은 자연어 처리를 통하여 BERT Q&A 동작 시간을 줄이기 위해서 각각 단어에 맞는 조사를 포함하여 추출한다. 예를 들어 “학생”이라는 주어 단어와 “들은”이라는 단어를 합쳤을 때 띄어쓰기 단어에 “학생들은”이라는 단어가 존재하므로 “학생들은”이라는 단어는 추출 주어가 된다. 각 추출 단어들은 이러한 자연어 처리 과정을 통해서 문장 내에 존재하는 단어로만 구성된다.

4.4 자연어 처리 결과의 질문 사항 조합을 통한 BERT Q&A 수행

이전 단계에서 수행한 단어 추출 단어들을 조합하여서 BERT Q&A의 질의를 만들어낸다. 예를 들어 “학생은 무엇을 볼 수 있는가?”에 대한 질문은 이전 단계에서 추출된 주어인 “학생은”과 추출된 질문 행동인 “볼 수 있는가?”와 의문사인 “무엇”을 조합하여 만들어 낸다. 이렇게 만들어진 문장을 BERT Q&A에 입력하게 되면 BERT Q&A는 질문에 대한 결과로 문장 내에서 “성적표를”이라는 결과를 출력하고 자연어 처리를 통해 해당 단어의 조사를 제외하여 “성적표”라는 단어를 저장한다. 이러한 방법으로 모든 추출 주어와 추출 질문 행동에 대해 조합하여 질의를 만들어 수행하고 결과를 저장하게 된다.

4.5 BERT Q&A 결과를 통한 온톨로지 구성

BERT Q&A를 통한 응답 결과를 자연어 처리하여 온톨로지를 구성한다. 예를 들면 각 응답에서 주어로 사용된 “학생”, “시스템”, “교수”, “학기”는 온톨로지의 식별자가 된다. 질문 문장인 “볼 수 있는가?”와 응답 단어인 “성적표”를 자연어 처리하여 “성적표 보기”로 단어를 생성하고 이를 식별자의 관계어로 구성한다. 이를 통해서 “학생”이라는 식별자에 “성적표 보기”라는 관계어가 형성된다. 이러한 방법으로 모든 주어와 응답 단어에 대해 조합하여 온톨로지를 구성한다.

4.6 RE Chatbot 응답을 통한 요구사항 문서 작성

위의 온톨로지를 베이스 데이터로하여 챗봇 Q&A를 수행한다. 예를 들면 챗봇은 “학생”, “시스템”, “교수”, “학기”라는 액터 후보군을 보여줄 것이고 사용자는 이 리스트 중에 요구사항에 맞는 액터를 선택한다. 다음으로 챗봇은 “가르칠 코스 등록”, “액세스 업데이트”, “성적표 보기”, “코스 등록” 등등의 행동 리스트 후보군을 보여줄 것이고 사용자는 이중 요구사항에 맞는 행동 리스트를 선택한다. 액터와 행동 선택이 종료되면 챗봇은 선택된 액터와 행동 리스트를 조합하여 시스템 이벤트 리스트를 구성하고 최종적으로 사용자에게 해당 시스템 이벤트 리

Table 1. Comparison with Other Processes

Table Head	Consistency	Understanding	Coverage	Human work
Baek [proposed method]	✓			✓
Nguyen [2]		✓		✓
Osman [3]			✓	✓
Wang [4]	✓			✓
Surana [5]			✓	✓
Panichella [6]		✓		✓
Akay [7]	✓		✓	

스트를 보여준다. 예를 들면 “학생”과 “성적표 보기”를 선택한 경우 챗봇은 이 액터와 행동을 조합하고 사용자에게 “학생이 성적표를 본다”라는 이벤트 리스트를 제공한다. 사용자는 챗봇이 제공하는 이벤트 리스트를 본 후 요구사항에 맞는 이벤트 리스트를 선택한다. 모든 과정을 수행하게 되면 챗봇은 사용자가 선택한 이벤트 리스트를 기반으로 하여 간단한 형태의 요구사항 문서를 만들게 된다.

4.7 요구 분석 프로세스 비교

Table 1을 통한 기존 연구와의 일관성, 이해도, 커버리지, 인적 작업 항목 비교를 통해서 제시한 프로세스의 성능을 나타내었다.

5. 결론 및 향후 연구 방향

한글 요구사항 문서에 대해 BERT 알고리즘을 적용하여 액터 및 행동을 도출하고 챗봇을 이용한 사용자와의 질의 응답을 통해서 요구사항 명세서 자동 생성에 대한 연구를 수행하였다. 연구를 통해 요구사항 명세서 문서에 대해서 형태소 분석을 수행하였고 이를 통해 BERT Q&A를 통한 응답 결과를 얻게 되었다. 이를 통해 요구사항 문서에서의 액터 및 행동을 추출하였다. 그리고 이 데이터를 챗봇을 활용하여서 사용자로 하여금 질의 응답을 통해 요구사항에 대한 선택을 최종적으로 선택하게 하였다. 이를 통해서 선택된 결과를 요구사항 명세서로 작성하여서 사용자에게 제공하였다.

아직 요구사항 명세서의 내용에서 가장 중요한 Event flow를 채우지 못하는 문제점이 있으나 향후 연구를 통해서 Event flow를 채울 수 있는 방법을 연구할 것이다.

감사의 글

이 논문은 2023년도 정부(과학기술정보통신부)의 지원

으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2021-0-00177).

참고문헌

1. Young-Yun Baek, Yong-Beom Park. (2022). Suggested automatic creation of use case diagrams through machine learning analysis in Korean requirements documents. *Journal of the Institute of Electronics Engineers of Korea*, 2737-2739.
2. Nguyen, Tuong Huan, John Grundy, and Mohamed Almosry. "Rule-based extraction of goal-use case models from text." *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. 2015.
3. Osman, Mohd Hafeez, and Mohd Firdaus Zaharin. "Ambiguous software requirement specification detection: An automated approach." *2018 IEEE/ACM 5th International Workshop on Requirements Engineering and Testing (RET)*. IEEE, 2018.
4. Wang, Yingxu, and James Y. Xu. "Autonomous Software Requirement Specifications towards AI Programming." *2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE, 2021.
5. Tlusty, J., Smith, S., and Zamudia, C., "Critical Reviews in Solid State and materials Sciences," *J. of The Korean Society of Semiconductor & Display Technology*, Vol. 39, pp. 51-52, 2008.
6. Panichella, Sebastiano, and Marcela Ruiz. "Requirements-collector: automating requirements specification from elicitation sessions and user feedback." *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020.
7. Akay, Haluk, and Sang-Gook Kim. "Reading functional requirements using machine learning-based language processing." *CIRP Annals* 70.1 (2021): 139-142.
8. Byung-Sun Cho, Seok-Won Lee. (2020). A Comparative Study on Requirements Analysis Techniques using Natural Language Processing and Machine

- Learning. *Journal of the Korea Society of Computer and Information*, 25(7), 27-37.
9. Abdouli, Mariem, Wahiba Ben Abdesslem Karaa, and Henda Ben Ghezala. "Survey of works that transform requirements into UML diagrams." 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA). IEEE, 2016.
 10. Liu, Kaihua, Sandeep Reddivari, and Kalyan Reddivari. "Artificial Intelligence in Software Requirements Engineering: State-of-the-Art." 2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI). IEEE, 2022.
 11. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
 12. Y. Lee and Y. Yu, mecab-ko-dic, <https://bitbucket.org/eunjeon/mecab-ko-dic>(accessed 26 Jan 2023)
-
- 접수일: 2023년 3월 6일, 심사일: 2023년 3월 16일,
게재확정일: 2023년 3월 16일