

플랜트 설비 문서로부터 설비사양 추출 및 유사설비 사양 교차 검증 접근법⁺

(A Method for Extracting Equipment Specifications from Plant Documents and Cross-Validation Approach with Similar Equipment Specifications)

이 재 현^{1)*}, 최 승 언²⁾, 서 효 원³⁾

(Jae Hyun Lee, Seungeon Choi, and Hyo Won Suh)

요 약 플랜트 엔지니어링 기업은 서로 다른 공종별 부서에서 플랜트 공정/설비/파이프/계장 등 각 관련 분야의 요구사항 문서를 작성하거나 참조하게 된다. 공정 관련 요구사항 문서는 공정에 대한 설명과 함께 이를 운영할 설비 또는 관련 시설의 요구사항들을 포함한다. 각 공종별 문서에 기술된 설비 또는 부품에 관련된 요구사항과 사양 정보는 문서의 작성자와 검토자들이 다르기 때문에 상호 간에 불일치가 발생할 가능성이 있다. 이 사항들에 대한 일치성을 확인하는 것은 전체 플랜트 설계 정보의 신뢰도를 높일 수 있다. 하지만, 문서의 양이 방대하고 서로 다른 문서들에 동일한 설비 부품에 대한 요구사항들이 일반 문장 형태로 흩어져 있기에 이를 사람이 추적하여 관리하는 것은 한계가 있다. 본 논문에서는 서로 다른 문서들 내에 기술된 요구사항 문장들을 분석하여 설비 또는 설비 부품과 관련된 요구사항 문장의 유사도를 계산하여 의미적으로 동일한 문장을 찾아내는 방법을 제안한다. 요구사항 문장의 유사도를 계산하기 위하여 의미적으로 요구사항의 중심이 되는 부품과 속성을 개체명 인식 방법을 활용하여 찾아내고, 찾아진 부품과 속성들의 유사도를 계산하여 두 문장이 의미적으로 동일함을 판단하는 방법을 제안한다. 플랜트 현장에서 사용하는 문서의 문장들을 예제로 하여 제안하는 방법을 설명하고 실험 결과를 설명한다.

핵심주제어: 플랜트 요구사항 문장 분석, 문장 내 엔터티 인식, 규칙 기반 추론

Abstract Plant engineering companies create or refer to requirements documents for each related field, such as plant process/equipment/piping/instrumentation, in different engineering departments. The process-related requirements document includes not only a description of the process but also the requirements of the equipment or related facilities that will operate it. Since the authors and reviewers of the requirements documents are different, there is a possibility that

* Corresponding Author: jaehyun.lee@daegu.ac.kr

+ 이 논문은 산업통상자원부 ‘소비재 제품 고객평가 데이터 AI 분석 및 제조 활용 서비스 개발’ (Project No: 20009185), 국토교통부 ‘AI기반 가스·오일 플랜트 운영·유지관리 핵심기술 개발’ (Project No: 21ATOG-C161933-01), 산업통상자원부 ‘화학플랜트 수직형 통합 스마트팩토리 패키지 개발’ (Project

No: 20009324) 프로젝트에 의해 지원되었음.

Manuscript received March 20, 2024 / revised April 05, 2024 / accepted April 11, 2024

1) 대구대학교 기계자동차공학부, 제1저자, 교신저자
2) 한양대학교 인공지능융합학과, 공저자
3) 한국과학기술원 산업및시스템공학과, 공저자

inconsistencies may occur between equipment or parts design specifications described in different requirement documents. Ensuring consistency in these matters can increase the reliability of the overall plant design information. However, the amount of documents and the scattered nature of requirements for a same equipment and parts across different documents make it challenging for engineers to trace and manage requirements. This paper proposes a method to analyze requirement sentences and calculate the similarity of requirement sentences in order to identify semantically identical sentences. To calculate the similarity of requirement sentences, we propose a named entity recognition method to identify compound words for the parts and properties that are semantically central to the requirements. A method to calculate the similarity of the identified compound words for parts and properties is also proposed. The proposed method is explained using sentences in practical documents, and experimental results are described.

Keywords: Plant requirement sentence, Named entity recognition, Rule-based reasoning.

1. 서론

플랜트 산업에서 발주사와 플랜트 설계사, 시공사 간에 의사소통의 핵심은 발주사의 플랜트 관련 요구사항이 정확히 전달되는 것이다. 플랜트 요구사항에는 플랜트에 설치되어야 하는 설비에 대한 사양 관련 요구사항만이 아니라 플랜트 설비가 운영되는 환경과 운영 공정을 고려한 요구사항들도 포함된다(Lee and Suh (2015)). 플랜트의 설계사와 시공사는 플랜트 발주사의 요구사항을 충족시키기 위해 플랜트 설계와 시공을 수행하기 때문에, 플랜트 요구사항 문서는 플랜트 설계 단계와 시공 단계를 위한 중요한 기준 입력 데이터가 된다. 플랜트의 다양한 공종들(설비, 배관, 구조물, 계측) 측면에서 각각의 요구사항들이 요구사항 문서에 담기게 되는데, 플랜트 요구사항 내용은 일반적으로 ITB(Invitation to bidder)라고 불리는 문서에 기술된다. 플랜트 설계와 시공 능력을 갖춘 엔지니어링 회사는 ITB 문서에 포함된 플랜트 설계 요구사항 및 설치/운영 요구사항들을 파악하여 플랜트 프로젝트를 수행하게 된다.

플랜트 설계와 관련된 요구사항 문서에는 엔지니어가 설비나 부품 설계에 고려해야만 하는 많은 요구사항 관련 문장들이 기술되어 있다. 이 요구사항 문장들이 실제 설계에 반영되었는지를 확인하기 위해서는 각 요구사항이 아이템화되어 설계 데이터와 매칭될 수 있어야 한다.

이를 통해 요구사항들이 설계 데이터에 반영되었는지를 검증할 수 있고, 요구사항 만족 여부를 엔지니어가 판단할 수 있다.

요구사항 문장을 아이템화한다는 것은 요구사항 문장이 설명하고자 하는 부품이 무엇이고, 그 부품과 관련된 속성, 그리고 속성값의 범위에 대한 설명이 명확히 인식되는 것을 뜻한다. 예를 들어, 다음과 같은 요구사항 문장이 있다고 가정하자. “Maximum operating speed of vertically suspended pumps shall not exceed 1,800 RPM unless approved by the Standards Committee Chairman.” 이 문장에서 부품은 ‘vertically suspended pumps’라 할 수 있고, ‘maximum operating speed’는 속성이라 할 수 있고, ‘not exceed 1,800 RPM’은 속성값이라 할 수 있다. ‘unless approved...’ 구는 조건문으로 요구사항이 적용되는 조건을 설명하고 있다.

플랜트 발주사의 요구사항 문장이 아이템화되지 않은 텍스트 형태로 주어지기 때문에 엔지니어는 이 문장들을 수작업으로 아이템화하여 설계 결과와 비교해야 하는 어려움이 있다. 적은 양의 요구사항을 비교한다면 문제가 없지만, 플랜트와 같이 하나의 설비에 다수의 요구사항이 존재하고 설비에 연결되는 다양한 파이프, 계측기, 구조물 등의 요구사항들도 함께 고려하기 위해서는 플랜트 요구사항 문장에 대한 아이템화가 필요하다.

일반적으로 텍스트 문장에서 특정한 클래스의

단어들을 추출하는 접근방법을 Named Entity Recognition(NER)이라 한다(Li et al., 2020). NER 접근방법에는 다양한 머신러닝 기법들이 활용되었는데, 대표적으로 Hidden Markov Model (Morwal et al., 2012), Conditional Random Field(Patil et al., 2020)와 같은 지도학습 접근방법이 있고, 대형 언어 모델을 활용하는 딥러닝 접근방법(Kalyan, 2024)이 있다. 각각의 접근방법이 장단점을 가지고 있다. 딥러닝 접근방법은 딥러닝 기술의 발전에 따라 엔지니어의 개입 없이 NER 수행의 정확도가 점점 더 높아지는 장점이 있으나, 기업의 핵심 데이터가 외부 서비스에 유출되는 문제를 갖고 있다. 지도학습 접근방법의 경우 엔지니어가 모델 학습 특징(features)을 어떻게 정의하는가와 학습을 위한 정답 데이터 구축 등 엔지니어의 개입이 많이 필요하지만, 데이터 보안 우려에서 자유로운 장점이 있다. 본 논문에서는 기업의 플랜트 요구사항 문서 내용을 적용하기 위하여 문서 내 일부 샘플 문장들을 활용하였고, 지도학습 접근방법과 딥러닝 접근방법을 비교하여 NER을 수행하였다.

요구사항 문장을 아이템화하기 위해 NER을 활용하지만, NER만으로는 요구사항 문장으로부터 적절한 부품, 속성, 속성값을 추출하기에는 한계가 존재한다. 예를 들어, “For vertically suspended pumps having driver rating of 220 kW (300 HP) or above and are not equipped with thrust bearings, two seismic sensors shall be provided on the thrust bearing housing of the driver”와 같은 요구사항 문장이 있다고 가정할 때 요구사항과 관련된 부품은 ‘two seismic sensors’가 되어야 하지만 NER 접근방법으로는 ‘vertically suspended pumps’도 부품으로 추출되기 때문이다. 이와 같이 중복적으로 엔터티가 찾아졌을 때 후처리 방법으로 본 논문에서는 규칙 기반 후처리 방법을 제안한다.

요구사항 문장이 적절히 아이템화된다면, 서로 다른 공종에서 정의되는 동일한 부품에 대한 요구사항 문장들이 동일한 내용을 요구하는지 검증이 가능해진다. 이는 두 개 이상의 요구사항 문장을 비교할 때, 문장의 형식적인 패턴이나

문법이 아니라 문장이 표현하고자 하는 요구사항 내용의 유사도를 계산할 수 있게 해준다. 본 논문에서는 아이템화된 요구사항 문장들의 유사도를 비교하는 방법으로 단어의 유사도 비교와 함께 부품, 속성에 대한 어휘체계(taxonomy) 구조를 활용한 유사도를 계산 방법을 제안한다.

본 논문에서 제안하는 방법은 자연어로 작성된 요구사항 문서로부터 설계 요구사항을 정형화된 형식으로 추출하고, 요구사항들 간 유사도를 계산하는 방법을 제안함으로써 궁극적으로 엔지니어가 요구사항 문서에 관한 내용 검토와 설계 사양 검증 과정을 지원하는 서비스 개발이 용이하게 한다.

2. 기존 연구

요구사항 문서 내 유사한 문장을 찾기 위해서는 문장 내 부품, 속성, 속성값에 해당하는 단어들을 찾아내고, 서로 다른 문장에 존재하는 이들 단어의 유사도를 비교할 수 있어야 한다. 이와 관련된 기존 연구를 살펴보면 다음과 같다.

2.1 개체명 인식

개체명 인식(NER: Named Entity Recognition) 연구는 문장에서 특정한 의미를 갖는 단어를 찾아내는 자연어 처리 방법을 뜻한다 (Telecommunications Technology Association, 2024). 본 논문에서는 요구사항 문장에서 Part, Property, Value에 해당하는 복합단어를 추출하기 위하여 개체명 인식 기법을 사용한다. Li et al. (2020) 연구에 따르면 개체명 인식 접근방법으로는 규칙 기반으로 개체명을 인식하는 방법, 주어진 텍스트 데이터를 기반으로 통계적인 방법을 적용하여 개체명을 인식하는 비지도 학습 방법, 그리고 개체명에 대해 라벨을 입력한 정답지를 기반으로 개체명 인식 모델을 만드는 지도 학습 방법이 있다.

개체명 인식은 문장의 감성분석 분야에서도 많이 활용된다. Woo et al. (2022)는 제품에 대한 한글 리뷰 문장에 대한 감성분석을 통해 제

품설계에 반영하는 방법을 연구하면서 양방향 LSTM과 조건부 무작위장(CRF: Conditional Random Field) 알고리즘을 활용하였다. Kim (2023)은 한글 문장의 감성분석을 위해 구글의 BERT 모델(Devlin et al. 2019)을 활용하여 미세조정을 하는 방법을 제안하였다.

본 논문에서는 요구사항 문서의 Part, Property, Value 개체를 인식하기 위해 BERT 모델의 경량화 버전인 ALBERT 모델(Lan et al., 2020)을 활용한다. ALBERT 모델은 BERT 모델의 경량화 버전이면서도 1.7배 더 빠르게 학습하고 좋은 성능을 보여준다.

2.2 단어 유사도 계산 방법

두 단어의 유사도를 계산하는 방법으로는 전통적으로 코사인 유사도, 유클리디안 거리 유사도, 자카드 유사도 등이 대표적인 방법이다. 자카드 유사도는 두 단어에 포함된 단어 집합의 교집합과 합집합 크기를 계산하여 유사도를 계산하는 방법이다. 코사인 유사도와 유클리디안 거리 유사도는 두 단어를 벡터 형태로 변환한 후 코사인 각도와 유클리디안 거리를 계산하여 유사도를 도출한다. 단어를 벡터로 변환하는 방법에는 워드 임베딩 기법(Mikolov et al., 2013)

을 활용한 방법과 BERT와 같이 사전 학습된 언어 모델을 활용한 방법이 존재한다.

두 문장의 유사도를 계산하는 방법도 존재하는데, 문장의 유사도 계산 방법은 문법 구조와 문장의 의미를 고려하기 때문에 단어의 유사도보다 복잡하게 된다. 대표적으로 문단을 벡터화(Le and Mikolov, 2014) 하여 코사인 유사도나 유클리디안 거리 유사도를 계산하는 방법이 있고, 문장의 문법적 구조를 분석하여 구조의 유사도를 도출하는 방법(Kong et al., 2014)이 있다. 또한, BERT와 같은 사전 학습 언어모델을 활용하여 문장의 의미적 유사도를 측정하는 방법이 있다.

본 논문에서는 두 요구사항 문장의 유사도를 측정할 때 각 문장의 주요 Part, Property, Value를 표현하는 복합단어들의 유사도를 측정하여 문장 유사도를 계산하는 방법을 제안한다. 복합단어의 유사도는 BERT와 같이 사전 학습된 언어 모델을 이용하는 것이 더 정확할 수 있지만, 엔지니어링 도메인에서 두 단어의 의미적 일치를 판단하기 위한 유사도 한계치를 결정하기에는 어려움이 있다. 이에 본 논문에서는 사전에 정의된 용어분류도(taxonomy)를 제공하여 용어분류도 구조를 활용한 단어 유사도 계산 방법을 제안한다.

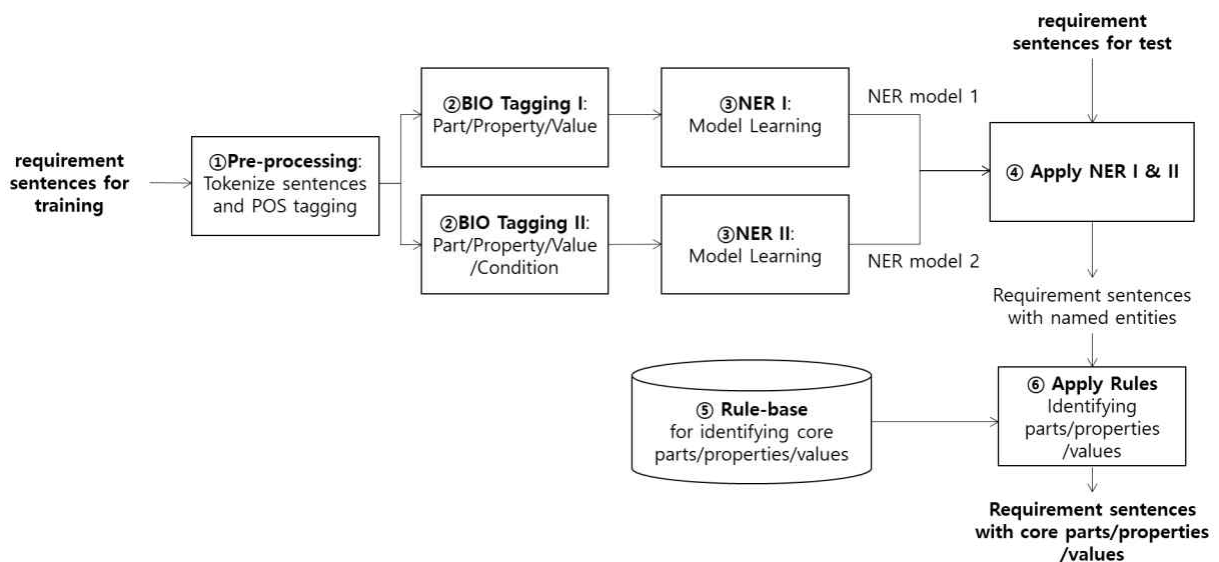


Fig. 1 The proposed 'two-way NER' approach for identifying named entities in requirement sentences.

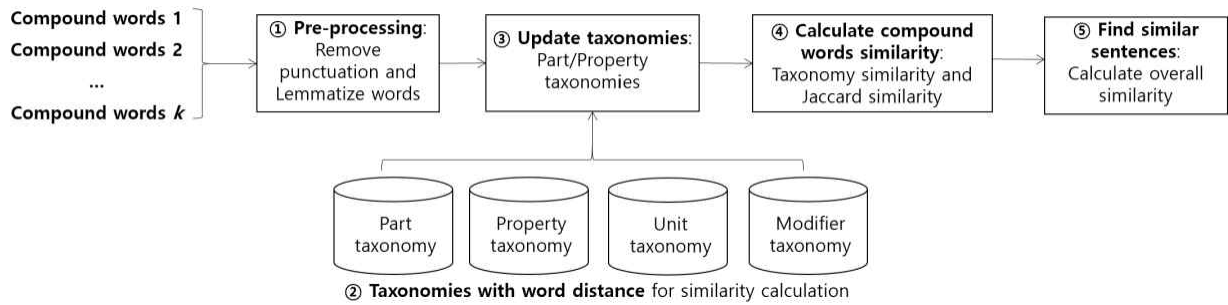


Fig. 2 Process to find similar requirement sentences with itemized part, property, and value words

3. 접근방법

3.1 Two-way NER 접근방법

요구사항 문장에서 부품(Part), 속성(Property), 속성값(Value)을 추출하기 위한 NER 접근방법을 직접적으로 적용하였을 때 문장 내 중심 부품과 중심 속성, 그와 관련된 속성값을 정확히 찾아내는 것에 한계가 있다. 이는 요구사항 문장에서 중심 부품은 문장의 주어부에 위치하게 되는데 문장의 조건절에도 관련 부품이 기술되기 때문이다. 따라서, 문장에서 주어부와 술어부, 조건부를 구분할 수 있으면 이를 활용하여 중심 부품과 중심 속성을 찾는 데 도움이 될 것이다. 문장에서 주어부와 술어부, 조건부를 찾아내는 것 또한 NER 접근방법을 통해 가능하다. 본 논문에서는 요구사항의 중심 부품과 중심 속성을 찾아내기 위해 2개의 NER 접근방법을 함께 활용하는 ‘Two-way NER’ 접근방법을 제안한다. Fig. 1은 제안하는 접근방법에 대하여 도식으로 설명한다.

Fig. 1에서 ①과 ②는 요구사항 문장에 대한 전처리 과정과 BIO 태깅 과정이다. 요구사항 문서로부터 추출된 문장들을 전처리하는 것은 문장들에 대해 토큰화 처리를 하고, 토큰화된 단어들에 대하여 Position of Speech(POS) 태그를 붙이는 것을 뜻한다. 영문에 대한 Tokenizer와 POS tagger는 Stanford NLTK 모듈을 활용한다. BIO 태깅은 2개의 NER 모델학습을 위한 훈련 데이터를 만드는 과정이다. Part, Property, Value를 추출하는 NER 학습 모델과 주어부

(Part), 술어부(Property, Value), 조건부(Condition)를 추출하는 NER 학습모델을 만들기 위한 훈련 데이터를 만든다. 본 논문에서는 1080개의 요구사항 문장에 대하여 엔지니어 전문가가 훈련 데이터를 만들어 모델 학습에 활용하였다. Fig. 1의 ③과 ④는 NER을 수행하는 프로세스로 각각 학습과 테스트를 수행한다. 본 논문에서는 NER 방법들 중 CRF방법과 Machine Reading Comprehension (MRC) 방법을 비교하였고, 둘 중 우수한 성능을 보이는 MRC 방법을 채택하였다. Fig. 1의 ⑤과 ⑥은 2 개의 NER 처리 결과를 활용하여 요구사항 문장의 중심 Part, Property, Value를 찾아내는 규칙들과 규칙적용 프로세스를 표현한다. 4장에서 NER 프로세스 비교와 규칙 처리에 대하여 상세히 설명한다.

3.2 요구사항 문장 유사도 계산 방법

Two-way NER 접근방법을 통해 요구사항 문장들의 중심 Part, Property, Value들이 아이템화되었다면 요구사항 문장의 유사도를 아이템화된 요소들 중심으로 계산이 가능하다. 요구사항 문장이 서로 다르게 표현되었다 하더라도 중심 Part, Property, Value의 유사도를 각각 계산하여 이를 종합하면 요구사항 문장의 유사도를 계산할 수 있게 된다. Fig. 2는 요구사항 문장들의 유사도를 아이템화된 Part, Property, Value 복합단어들의 유사도 계산을 통해 구하는 프로세스를 보여주고 있다.

Table 1 Meaning of entities in the mid and small classification

Entity	Mid classification	Small classification
Part	Subject	Equipments or parts
Property	Verb	Properties
Value	Complement	Required Values
Condition	Condition	-

중심 Part, Property, Value 단어들은 한 개의 단어로 구성되기보다는 여러 개의 단어로 구성된 복합단어로 표현되는 경우가 많다. 복합단어의 유사도를 계산하기 위해서 본 논문에서는 Part, Property, Unit, Modifier 용어들에 대한 용어분류도를 활용한다. 용어분류도는 트리 구조로 용어들이 분류되어 있으며 해당 도메인에서 활용하는 대표적인 용어들이 포함된다. Value 복합단어의 경우 일반적으로 특정한 수치를 표현하는 숫자를 중심으로 앞부분에는 ‘크다’와 ‘작다’ 등을 표현하는 용어들(Modifier)이 오고, 숫자 뒤에는 물리량을 표현하는 단위(Unit)이 표현된다. Value 복합단어의 유사도는 숫자의 동일 유무와 함께 Modifier와 Unit의 유사도를 함께 계산한다.

복합단어의 유사도를 계산하는 방법과 이를 통해 요구사항 문장의 유사도를 계산하는 방법은 5장에서 상세히 설명한다.

4. Two-way NER 접근방법

4.1 BIO 태깅과 NER 모델 학습

플랜트 ITB 문서에 기술된 요구사항 문장을 보면 몇 가지 특징을 찾을 수 있다. 모든 문장이 특정 요구사항을 표현하기 때문에 주어부가 명확한 문장으로 구성되고 주어부는 1개로 구성된 문장들이다. 요구사항이 특정 환경이나 조건에 적용되는 경우들이 있기 때문에 문장에 조건을 표현하는 부분이 포함되기도 한다. 플랜트의 특정 부품이나 설비에 대한 요구사항이기 때문에 주어부에는 부품이 자주 언급된다. 속성은

주어부에 부품과 함께 표현되기도 하고 술어부에 속성값과 함께 표현되기도 한다. 또한, 속성이 주어부와 술어부에 없는 경우에는 조건부에 언급되는 경우도 있다. 속성값의 경우에는 수치가 명확히 표현되는 경우도 있고, 수치가 없는 경우도 있다. 수치가 있는 경우에는 그 숫자의 물리량을 표현하는 단위가 표현되고 수치의 범위를 표현하는 경우도 존재한다. 하나의 문장 내에 여러 부품과 속성, 속성값이 표현되는 경우도 있는데, 부품과 속성을 표현하는 단어 간 문장 내 위치가 가까울수록 상호 관련성이 높다. 가끔씩은 문장에서 부품의 속성이 생략되는 경우도 있는데, 이는 앞 문장에 언급되어서 생략되거나, 속성값의 단위나 조건부 내용을 통해 유추가 가능하다.

Two-way NER 접근방법은 크게 전처리, BIO 태깅, NER 모델 학습과 신규 문장에 대한 NER 처리, 그리고 규칙 적용 단계로 구분할 수 있다.

전처리: 전처리는 요구사항 문장들에 대하여 토큰화(Tokenize)하는 과정과 소문자 처리, 문장 부호 처리를 수행한다.

BIO 태깅: 두 개의 NER 모델 학습과 적용에는 Part, Property, Value에 대한 BIO 태그를 학습하는 것과 문장 구조를 고려한 Part, Property, Value, Condition에 대한 BIO 태그를 학습하는 것이 진행된다. 본 논문에서는 편의상 첫 번째 학습을 소분류 학습이라 하고, 두 번째 학습을 중분류 학습이라 부르겠다.

BIO 태깅은 소분류 학습을 위한 태그와 중분류 학습을 위한 태그를 입력한 2 개의 파일을 만든다. 소분류와 중분류 태그를 통해 인식해야 하는 엔터티는 Table 1과 같다.

이와 같은 엔터티 정의를 활용하여 하나의 문장에 대하여 서로 다른 두 개의 BIO 태그 정보가 정의된다. 예를 들어, “The rated speed of vertical suspended pumps shall not exceed 1,800 RPM unless approved by the Standards Committee Chairman”이라는 문장에 대하여 BIO 태그를 하면 Table 2와 같다.

NER 모델 학습: NER 모델 학습 알고리즘들의 성능을 비교한 연구(Li et al., 2020)를 보면 과거 HMM, CRF 알고리즘의 성능보다 BERT

Table 2 A BIO tagging example for the two-way NER approach

Token	Mid Class	Small Class
The	B-Part	O
rated	I-Part	B-Property
speed	I-Part	I-Property
of	I-Part	O
vertical	I-Part	B-Part
suspended	I-Part	I-Part
pumps	I-Part	I-Part
shall	B-Property	O
not	I-Property	B-Value
exceed	I-Property	I-Value
1,800	B-Value	I-Value
RPM	I-Value	I-Value
unless	B-Condition	O
approved	I-Condition	O
by	I-Condition	O
the	I-Condition	O
...

알고리즘을 적용했을 때 더 우수한 성능이 나왔다. 요구사항 문장에 대해서도 동일한 결과인지 확인을 위해 1,080개의 샘플 문장들에 대해 비교 실험을 진행했다. 소분류의 경우 CRF 알고리즘의 f1 score 값이 최대 0.656 나왔고 ALBERT 알고리즘을 적용한 경우 f1 score 값이 최대 0.923이 도출되었다. 본 논문의 NER 모델 학습에 ALBERT 알고리즘을 활용한다.

ALBERT는 BERT의 경량화 버전으로 NER에 우수한 성능을 내는 것으로 알려져 있다. 본 논문에서는 ALBERT 모델 중 파라미터가 가장 많은 albert_xxlargev1_squad2_512 모델(Lan et al., 2020)을 활용하였다. 위키피디아 문서를 바탕으로 사전 학습된 ALBERT 모델에 240개의 요구사항 문장들로 미세조정 학습을 하여 중분류와 소분류 NER을 수행하도록 한다. ALBERT 모델의 미세조정 학습은 BIO 태그 데이터를 SQuAD 2.0 형식의 질의답변 데이터로 변환하여 수행한다. SQuAD 2.0 형식은 Stanford Question Answering Dataset의 확장 버전으로 문장/문서에 대한 질문과 그 질문에 대한 대답이 포함되어 있고, 만약 정답이 없다면 해당 내용이 없다는 답변도 표현할 수 있다. 예를 들어,

Table 2의 문장에서 소분류에 해당하는 Part를 찾으라는 질문과 답변을 SQuAD 2.0으로 표현하면 아래와 같다.

```
[{"title": "pump design SQuAD 2.0",
"paragraphs": [{"qas": [{"question": "What apparatus is being discussed, including modifiers?", "id": "0", "answers": [{"text": "vertical suspended pumps", "answer_start": 20}], "context": "The rated speed of vertical suspended pumps shall not... (omitted)"}]}
```

이렇게 학습한 중분류와 소분류 NER 모델을 신규 요구사항 문장들에 각각 적용하였을 때 각 모델별로 Part, Property, Value를 찾아내고 중분류 모델의 경우 Condition도 함께 찾아내게 된다.

4.2 Rule-base를 활용한 추론

Rule-based 추론은 각 문장에서 중분류 범위에 포함되는 Part, Property, Value로부터 소분류의 Part, Property, Value 결과를 활용하여 적절한 Part, Property, Value를 추론하는 직접적인 방법이다. Rule을 정의하는 방법은 크게 IF문과 THEN문으로 구성된다. Table 3은 본 논문에서 제안하는 방식의 Rule 정의 예제를 보여준다.

IF문과 THEN문에는 각각 문장 내 패턴을 찾아내는 패턴구와 실행을 수행하는 함수구가 AND 논리('&'로 표현됨)로 연결된다. 패턴구는 문장 내 추출된 단어의 엔터티 타입(Part, Property, Value)과 중분류/소분류 구분(m: 중분류, s:소분류)을 합쳐진 형태로 해당 단어의 패턴을 찾아낸다. 예를 들어, Rule ID 1번의 IF문의 "m:part(s:part)"는 소분류 Part에 정의된 단어들, 중분류도 정의된 단어들과 같은 것들을 찾게 된다. 다른 예로 Rule ID 2번의 IF문의 "m:property()"라는 표현은 중분류에 Property가 있지만, 소분류에 Property가 없는 패턴을 찾게 된다. 단어의 패턴을 명확히 찾아내기 위해 패턴구는 변수를 활용할 수 있다. 변수는 어떤 단

Table 3 Two-way NER applying rule examples

Rule ID	IF	THEN
1	m:part(s:part)	part(s:part)
2	m:property()&equal(m:property, 'shall')	property()
3	m:value(s:value)	value(s:value)
4	m:part(s:property+'of'+X_0)	property(s:property)&assign(X_0,part)
5	m:property()&m:condition(s:property)	property(s:property)

어든 해당 패턴에 매칭되는 단어를 표현한다. 예를 들어, Rule ID 4번의 IF문에서 m:part(s:property+'of'+X_0)라는 패턴구는 중분류 Part에 있는 단어들이 소분류 Property 단어와 'of' 단어, 그리고 이후 어떤 단어가 오든 'X_0' 변수에 저장될 수 있는 패턴을 찾아낸다. 매칭되는 패턴이 존재하면 True를 돌려주고, 그렇지 않으면 False를 돌려준다. 매칭이 될 경우 변수에는 해당 단어들이 임시로 저장된다.

함수구는 문장내 단어 패턴이외에 직접적으로 실행을 필요로 하는 부분을 정의한다. 예를 들어, equal(m:property, 'shall')과 같은 표현은 중분류 Property 단어가 'shall'과 같은지 여부를 체크하여 True/False를 돌려준다. THEN문에 활용되는 Rule ID 4번의 assign(X_0,part)와 같은 표현은 IF문에서 찾아낸 X_0 변수에 저장된 단어들을 그 문장의 Part로 지정하는 함수구이다.

Table 3의 규칙의 의미는 다음과 같다. 규칙 1번은 중분류 Part 단어들 중 소분류 Part 단어들과 동일한 것이 있다면 해당 단어들을 그 문장의 Part로 선언하는 규칙이다. 규칙 2번은 중분류 Property와 동일한 소분류 Property 단어들이 없고, 중분류에 'shall'과 동일한 단어가 Property로 정의되어 있다면 그 문장의 Property는 없는 것으로 선언하는 규칙이다. 규칙 3번은 중분류 value 단어들 중 소분류 Value 단어와 동일한 것이 있다면 이들을 그 문장의 Value로 선언하는 규칙이다. 규칙 4번은 중분류 Part 단어들 중 소분류의 Property 단어와 'of' 그리고 어떤 단어들이 존재한다면 이를 'X_0' 변수라 하고, 그 문장의 Part로 'X_0'에 저장된 단어를 선언하고, 소분류 Property를 그 문장의

Property로 선언하는 규칙이다. 규칙 5번은 중분류 Property에 매칭되는 소분류 Property가 없고 중분류 Condition에 매칭되는 소분류 Property가 있다면, 해당 소분류 Property를 그 문장의 Property로 선언하는 규칙이다.

Rule-base 추론의 장점은 중분류와 소분류 NER 결과물에 대하여 Part, Property, Value를 명확히 찾아낼 수 있다는 장점이 있다. 반면에 요구사항 문장의 형태가 복잡할 경우 Part, Property, Value 각각에 대해 여러 개의 단어들이 후보가 되는 경우도 발생하게 된다. 이를 제어하기 위해서는 규칙을 모두 적용하기 보다는 규칙 적용의 선후 관계를 정하는 방법과 모든 규칙 적용 후 후처리를 통해 필요없는 후보를 제거하는 과정이 필요하다. 본 논문에서는 Part, Property, Value에 다수의 후보가 발생할 경우 후처리를 적용하는 것을 우선적으로 고려하였다. 규칙적용 후처리에 대해서는 5장의 유사도 계산방법 설명에서 상세히 설명한다.

5. 복합단어 유사도 계산방법

플랜트 설계 관련 요구사항 문장이 서로 다른 공종에 기술되지만 동일한 설비나 부품에 대하여 기술되는 문장들이 존재한다. 이 문장들이 동일한 요구사항을 설명하는가에 대하여 확인하기 위해서는 요구사항 문장들을 비교하여 유사한 문장을 찾아내어야 한다. Fig. 3은 서로 다른 요구사항 문서에 존재하는 동일한 부품에 대한 요구사항을 설명하는 문장들을 찾아서 two-way NER 접근방법과 규칙들을 적용한 결과 예제를 보여주고 있다. 이들 문장들의 유사도를 측정하

Table 4 Examples of different requirement sentences for the same part and property

Sentence 1	Vertically suspended jockey pumps shall not exceed 1,800 RPM unless approved by the Standards Committee.		
Sentence 2	Maximum operating speed of vertically suspended pumps shall not exceed 1,800 RPM unless approved by the Standards Committee.		
Sentence 3	The rated speed of vertical suspended pumps shall not exceed 1,800 RPM unless approved by the Standards Committee.		
Sentence 4	For vertically suspended pumps having driver rating of 220 kW (300 HP) or above and are not equipped with thrust bearings, two seismic sensors shall be provided on the thrust bearing housing of the driver.		
Sentence ID	Part	Property	Value
1	Vertically suspended jockey pumps	-	not exceed 1,800 RPM
2	vertically suspended pumps	Maximum operating speed	not exceed 1,800 RPM
3	vertical suspended pumps	The rated speed	not exceed 1,800 RPM
4	Two seismic sensors	-	provided on the thrust bearing housing

는 방법은 문장들 간의 유사도를 직접 계산하는 방법과 요구사항 문장에서 핵심이 되는 Part, Property, Value의 유사도를 계산하여 문장들의 유사도를 계산하는 방법이 있다.

문장들 간의 유사도를 측정하는 방법은 문장 내 단어들을 사전에 학습한 BERT 모델을 활용하여 벡터화 하고 벡터들 간의 cosine 유사도를 계산하는 방법이 있다. 하지만, 두 문장에 유사한 단어가 있으면 유사도가 높아지는 경향이 있기에 내용상 유사하지 않은 문장들도 높은 유사도를 갖게 되는 단점이 있다. 예를 들어 Table. 4의 4개 문장의 cosine 유사도를 BERT 모델을 활용하여 계산하면 다음과 같다.

$$BERT_cos_sim(sent_1, sent_2) = 0.895$$

$$BERT_cos_sim(sent_1, sent_3) = 0.949$$

$$BERT_cos_sim(sent_2, sent_3) = 0.940$$

$$BERT_cos_sim(sent_1, sent_4) = 0.757$$

$$BERT_cos_sim(sent_2, sent_4) = 0.874$$

$$BERT_cos_sim(sent_3, sent_4) = 0.797$$

예제 문장에서 1, 2, 3번 문장은 내용이 유사하지만, 4번 문장은 이들과 다른 내용을 표현하고 있다. 하지만, 2번과 4번 문장의 유사도가 0.874로 비교적 높은 유사도를 갖는 것으로 계산된다. 이를 개선하기 위해 본 논문에서는 Part, Property, Value로 아이템화된 복합단어의 유사도를 계산하여 요구사항 문장의 유사도를

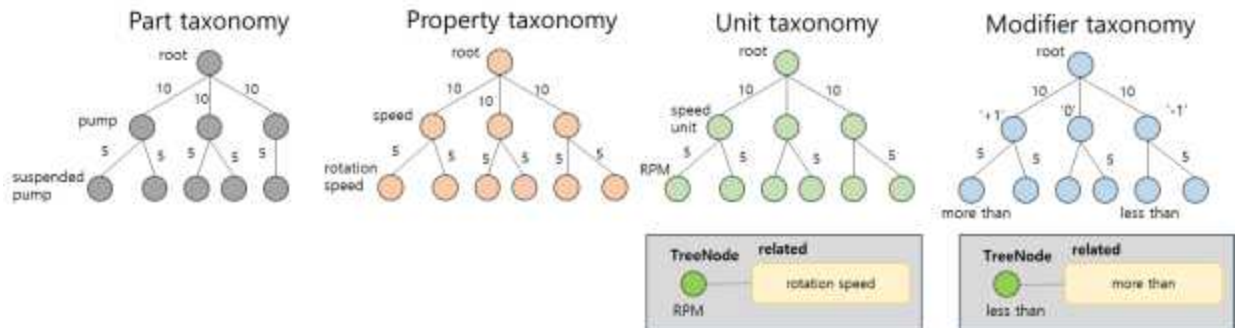


Fig. 3 Part, Property, Unit, Modifier taxonomy structure example

계산하는 방법을 제안한다. 이러한 접근은 요구사항 문장 간의 유사도가 왜 높게 나오는지에 대하여 명확한 이유를 찾을 수 있는 장점이 있다.

복합단어 유사도를 활용한 요구사항 문장 유사도 계산방법은 크게 전처리, 용어분류도 업데이트, 복합단어 유사도 계산, 문장 유사도 계산 단계로 이루어진다.

전처리: 복합단어 유사도 계산을 위한 전처리 과정은 소문자 처리, 관사와 문장부호 제거, 어간처리(lemmatization) 과정, 규칙적용 후처리를 거친다. 관사 및 문장부호 제거, 어간처리는 NLTK 패키지를 활용한다. 규칙적용 후처리는 규칙 적용 이후 Part 또는 Property에 적절한 단어가 찾아지지 않았을 때 적용되는 과정이다. 예를 들어, Table. 4의 규칙적용 이후 데이터 중 1번 문장의 결과를 보면 Property가 찾아지지 않고 Part와 Value만 찾아진 경우이다. Property가 찾아지지 않고 Value에 단위가 표현된 경우, 단위에 적절한 Property를 추정할 수 있다. 이와 같은 추정을 위해 사전에 정의된 Property 용어분류도와 Unit 용어분류도 간에 관계가 정의되어 있다. 'RPM'이라는 단위 용어는 'rotation speed'라는 Property 용어와 관계를 사전에 정의하였고, 이를 바탕으로 Property를 추정하게 된다.

Fig. 3는 Part, Property, Unit, Modifier 용어분류도의 구조를 표현하고 있다. 모든 용어분류도는 트리 구조를 가지고 있고, 부모-자식 트리노드들 간에는 노드 간 거리를 갖는다. 거리의 크기는 위로 갈수록 큰 값을 갖고 아래로 갈수록 작은 값을 갖도록 하되 최저값은 1로 정한다. 이 노드 간의 거리는 복합 단어들의 유사도를 계산하는데 활용된다. Unit 용어분류도의 경우 각 트리 노드는 Property 용어분류도의 노드와 하나의 'related' 관계를 갖는다. Modifier 용어분류도는 크다, 같다, 작다를 각각 '+1', '0', '-1' 노드로 매칭하여 용어분류도를 구성한다. Modifier 용어분류도의 트리 노드의 'related' 관계는 반대되는 개념의 Modifier 트리노드를 매칭한다.

용어분류도 업데이트: 용어분류도가 규칙적

용 후처리에도 활용되지만, 복합단어의 유사도 계산에도 사용된다. 요구사항 문장들로부터 아이템화된 모든 Part, Property 복합단어들이 사전에 정의한 Part와 Property 용어분류도에 존재하지 않을 수도 있다. 따라서, Part와 Property 용어분류도에 존재하지 않는 복합 단어들은 가장 유사한 용어를 찾아서 해당 용어분류도에 추가되어야 한다. Fig. 4는 Part 용어분류도에 신규 용어가 추가되는 모습을 도식적으로 보여주고 있다.

Fig. 4처럼 신규 용어가 추가되기 위해서는 기존 용어분류도의 용어들과 복합명사 유사도를 계산해야 한다. 보통 단어 유사도 계산 방법에는 자카드 유사도, 유클리디언 유사도, 코사인 유사도가 많이 활용된다. 하지만, 용어분류도의 복합단어의 경우 단어의 순서가 중요하다. 'pump'의 하위 단어에는 마지막에 'pump'가 오는 것이 일반적인 경우라 할 수 있다. 따라서, 신규 용어를 용어분류도의 트리노드와 비교할 때에는 마지막 단어가 동일한지 확인해야 하고, 동시에 단어의 순서가 얼만큼 비슷한지를 통해 유사도를 계산하는 것이 효과적이다. 단어의 순서 유사도를 계산하는 방법은 SequenceMatcher 알고리즘(Python, 2024)을 활용하였다. 신규 복합단어와 가장 유사도가 높은 트리노드를 찾으면 해당 노드의 부모 노드가 될지 형제 노드가 될지는 복합단어를 구성하는 각 단어들의 포함 여부를 확인하여 정하게 된다. 예를 들어, Fig. 4에서 어간 처리된 'vertical suspended pumps'와 'vertically suspended jockey pumps'는 각각 'vertic suspend pump'와 'vertic suspend jockey pump'로 변환되어 높은 유사도 값을 갖게 되는데, 신규로 추가되는 'vertical suspended pumps'의 단어들이 모두 포함되므로 'vertically suspended jockey pumps' 노드의 부모로 추가되게 된다. 만약 신규로 추가되려는 복합단어와 모든 트리노드의 단어들을 비교하였을 때 모든 유사도 값이 한계치(threshold)를 넘지 못할 경우 해당 신규 단어는 'root' 노드의 자식 노드로 추가된다.

복합단어 유사도 계산: 요구사항 문장에서 추출된 Part, Property 복합단어들이 모두 Part,

Property 용어분류도에 업데이트 되었기 때문에 서로 다른 문장의 Part와 Property들을 비교할 수 있다. Part와 Property 복합단어의 유사도를 계산하는 방법은 앞서 용어분류도 업데이트를 위해 용어 유사도 계산하는 방법들을 활용할 수도 있으나 본 논문에서는 taxonomy structure를 활용한 유사도 계산방법을 제안한다.

$$taxsim(w1, w2) = 1 - \frac{MinCut(w1, w2)}{MaxCut(w1, w2)} \quad (1)$$

taxonomy structure 기반 유사도(*taxsim*) 계산 방법은 두 용어(*w1, w2*)가 하나의 taxonomy에 존재한다고 가정하였을 때 트리 구조에서 최대 거리(*MaxCut*)와 최소 거리(*MinCut*) 계산을 통해 유사도를 계산한다.

Fig. 5은 용어분류도 구조기반 유사도 계산방법을 적용한 예제를 보여주고 있다. 해당 그림에서 두 용어의 최대 거리는 각 용어로부터 root노드까지 거리의 합이 되고, 최소 거리는 두 용어의 공통 부모 노드 중 제일 거리가 짧은 것까지의 거리 합이 된다. 따라서 *MaxCut(w2, w4)*는 최대거리 합인 32가 되고, *MinCut(w2, w4)*은 두 용어로부터 *w1*까지 거리의 합이기에 2가 된다.

두 복합단어 유사도가 일정한 한계점 이상이면 동일한 의미의 단어로 결정한다. 용어분류도의 트리 구조의 깊이(depth)가 일정 수준 이상일 때 그 수준보다 하위의 트리노드 단어들을 동일한 의미의 단어들로 간주할 수 있다면, 이를 반영하기 위한 한계점을 지정할 수 있다. 예를 들어 Fig. 5에서 ‘suspended pump’ 단어는

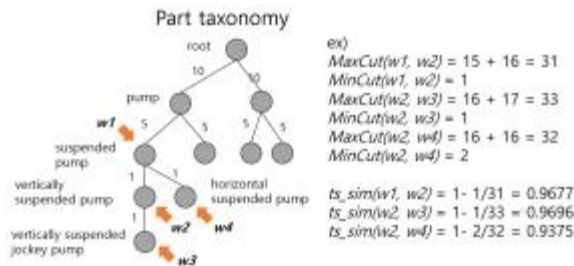


Fig. 5 Taxonomy structure-based similarity calculation example

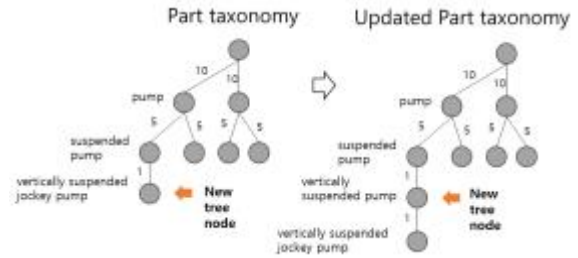


Fig. 4 Example of updating part taxonomy

Table 5 NER model f1 score evaluation

Domain	Part	Property	Value
Pump	0.923	0.758	0.837
Vessel	0.706	0.705	0.857
Tank	0.835	0.758	0.847

사전에 taxonomy 내에 정의된 단어이고, 이 단어 수준을 기점으로 자식 노드들(‘vertically suspended pump’와 ‘horizontally suspended pump’)은 서로 다른 것으로 하고, 자식 노드의 자손 노드들은 동일한 것으로 하고자 한다면 한계점을 0.968로 결정하면 된다.

용어분류도의 구조적 유사도는 트리구조 관점에서 유사도를 계산하기에 한계점을 사전에 정의가 가능하다는 장점이 있다. 만약 Fig. 5의 복합용어들에 대하여 BERT를 활용한 코사인 유사도를 계산하면 용어들 간의 유사도를 계산할 수 있으나 어느 수준의 한계점을 지정해야 동일한 용어로 간주할 수 있을지는 사전에 정의가 어려운 단점이 있다.

요구사항 문장에서 아이템화된 Value 복합단어들은 일반적으로 수치를 갖고 해당 수치의 범위를 표현하는 수식어(modifier)와 단위 단어를 포함한다. 유사한 요구사항 문장을 찾는데 있어서 Value 복합단어의 수치나 수식어, 단위가 항상 동일하다고 가정하기는 어렵기 때문에, Value 복합단어 간 유사도 계산 결과는 유사한 요구사항 문장을 찾는 과정에 옵션으로 적용될 수 있도록 한다. Value 복합단어의 유사도는 수치의 유사도와 수식어 유사도, 단위 유사도의 가중치 합계로 계산한다.

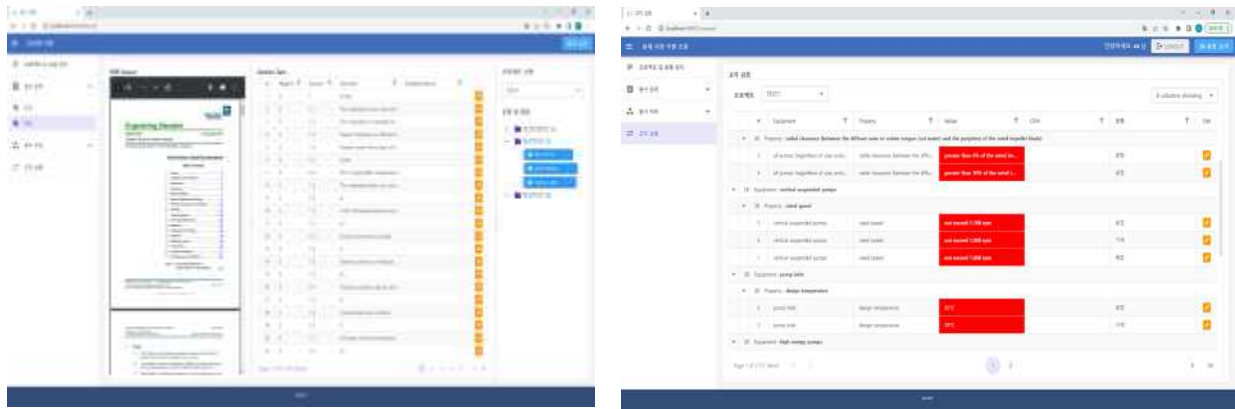


Fig. 6 Implementation screen-shots

6. 실험 결과와 구현

본 논문에서는 국내 플랜트 엔지니어링 기업의 Pump, Vessel, Tank 관련 ITB 문서로부터 문장을 샘플로 추출하여 학습 데이터를 만들었다. ALBERT를 활용한 NER 학습 모델을 만들기 위해 Pump, Vessel, Tank 도메인 별로 80개 문장으로 학습 데이터를 만들어 중분류와 소분류 NER 모델을 만들었다. 가장 좋은 성능을 보인 소분류 학습 모델의 테스트 평가 결과 f1 값은 Table 5와 같다. 중분류의 경우 테스트 평가 결과 f1 값이 0.922에서 0.989 사이 값들이 나왔다.

학습된 NER 모델들을 사용하여 유사한 문장을 잘 찾아내는지 평가하기 위해 Part, Property, Unit, Modifier 용어분류도를 수작업으로 만들고 40개의 테스트 문장을 생성하였다. 40개 테스트 문장은 ITB 요구사항 문서의 샘플 문장 중 Part, Property, Value가 포함된 문장을 12개 추출한 후 ChatGPT를 활용해 각 문장을 다른 형태로 변환한 문장들을 각 3~4개씩 만들었다. 이 테스트 문장들로 실험한 결과 precision: 0.897, recall: 0.614, f1 score: 0.729 평가 결과를 얻을 수 있었다.

유사한 문장을 찾는 부분에서 용어분류도에서 복합단어들의 동의어를 판단하는데 활용하는 한계점을 적절히 조절하면 이보다 더 좋은 결과를 얻을 수 있을 것으로 판단된다.

Fig. 6은 플랜트 ITB 문서 중 PDF 버전의 문서를 읽어들이어서 문장을 추출하고, 각 문장의 Part, Property, Value를 추출하여 유사한 문장들을 찾아내는 과정을 구현한 화면을 보여주고 있다. Fig. 6의 왼쪽 화면은 ITB문서에서 문장들을 추출하는 화면을 보여주고 있다. 오른쪽 화면은 문장들로부터 Part, Property, Value 엔티티를 추출하여 유사하다고 판단되는 문장들을 모아서 보여주는 화면이다.

사용자 화면은 웹페이지에서 작동하도록 구현하였고, NER 처리와 Rule 처리, 유사도 계산 알고리즘은 Python 버전 3.11로 구현하여 서버에서 실험되었다. 구현된 알고리즘들이 작동한 서버의 사양은 GPU(NVIDIA TITAN RTX), CPU(Intel(R) Core(TM) i9-10900KF CPU @ 3.70GHz), RAM(16GB)이었다.

7. 결론

플랜트 엔지니어링을 위한 중요 입력문서인 ITB에 포함된 요구사항 문장들은 구조적으로 문장이 간결한 문장들이 많으며 Part, Property, Value를 모두 포함하거나 일부만 포함하는 경우가 있다. 다양한 공종의 관점에서 동일한 설비나 부품에 대한 요구사항들이 서로 다른 문서에 기술되기 때문에 요구사항들이 상호 일치하

는지, 설계 결과가 요구사항들을 만족하는지 자동으로 확인이 필요하다. 본 논문에서는 이와 같은 요구사항 문장들의 유사도를 계산하기 위하여 요구사항 문장 내에 존재하는 Part, Property, Value 복합단어들을 인식하고, 이들 간의 유사도를 계산하는 방법을 제안하였다. 요구사항 문장의 특성상 단순히 Part, Property, Value 개체명을 인식하는 방법으로는 문장의 중심 Part, Property, Value를 찾아내기 어렵기 때문에 'two-way NER' 접근방법을 제안하였다. 'two-way NER'은 중심 Part를 찾아내기 위하여 문장에서 찾아진 Part들 중 주어부에서 우선적으로 찾을 수 있도록 2 단계의 NER 결과에 규칙을 적용하는 접근방법이다. 개체명 인식은 ALBERT 알고리즘을 바탕으로 지도학습을 수행하여 학습모델을 만들어 활용하였다.

기존의 유사도 계산방법과는 다르게 용어체계도의 구조적 유사도를 활용하여 복합용어들의 유사도를 계산하기 때문에 동일한 용어 유무를 판단하는 기준값을 각 용어들에 맞춰서 사전에 지정할 수 있기 때문에, 엔지니어가 요구사항 문서 내용과 설계 사양 검증 과정에서 동일한 부품, 속성을 판단하도록 지원하는 서비스 개발이 용이하고 설명이 가능한 장점을 갖는다. 다만, 도메인에 적절한 용어체계도를 사전에 개발하는데 많은 노력이 필요한데, 요구사항 문서들로부터 적절한 용어들을 수집하여 용어체계도를 자동으로 구축하기 위한 추가적인 연구가 필요하다.

References

- Devlin, J., Chang, M. W., K., L. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, May. 24.
- Kalyan, K. S. (2024). A Survey of GPT-3 Family Large Language Models including ChatGPT and GPT-4, *Natural Language Processing Journal* 6, 1-48.
- Kim, J. S. (2023). A Study on Fine-Tuning and Transfer Learning to Construct Binary Sentiment Classification Model in Korean Text, *Journal of Korea Society of Industrial Information Systems* 28(5), 15-30.
- Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C. and Smith, N. A. (2014). A Dependency Parser for Tweets, *2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *International Conference on Learning Representations*, <https://doi.org/10.48550/arXiv.1909.11942>.
- Le, Q. V. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *International Conference on Learning Representations*, <https://doi.org/10.48550/arXiv.1405.4053>.
- Li, J., Sun, A., Han, J. and Li, C. (2020). A Survey on Deep Learning for Named Entity Recognition, *IEEE Transactions on Knowledge and Data Engineering* 34(1), 50-70.
- Morwal, S., Jahan, N. and Chopra, D. (2012). Named Entity Recognition using Hidden Markov Model (HMM), *International Journal on Natural Language Computing* 1(4), 15-23.
- Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations*, <https://doi.org/10.48550/arXiv.1301.3781>.
- Python (2024). difflib - Helpers for Computing Deltas, <https://docs.python.org/3/library/difflib.html> (Accessed on March. 1, 2024)
- Telecommunications Technology Association

(2024). *Information and Communication Terminology* <http://terms.tta.or.kr> (Accessed on Mar. 1., 2024)

Woo, J. H., Jeong, M. K., Lee, J. H. and Suh, H. W. (2022). A Study of Customer Review Analysis for Product Development based on Korean Language Processing, *Journal of Korea Society of Industrial Information Systems* 27(1), 49-62.



이 재 현 (Jae Hyun Lee)

- 정회원
- 한국과학기술원 산업공학과 학사/석사/박사
- (현재) 대구대학교 공과대학 기계자동차공학부 부교수
- 관심분야: 제품설계정보시스템,

가상플랜트, 온톨로지 활용



최 승 언 (Seungeon Choi)

- 한양대학교 ERICA캠퍼스 응용수학과 학사
- (현재) 한양대학교 인공지능융합학과 석사
- 관심분야: 딥러닝, 텍스트 마

이닝, 강화학습



서 효 원 (Hyo Won Suh)

- 연세대학교 기계공학과 학사
- 한국과학기술원 기계공학과 석사
- West Virginia University 산업공학과 박사
- 대우중공업(주) 중앙연구소 주임연구원/현 '두산인프라코어'
- 한국생산기술연구원 생산시스템센터 수석연구원
- (현재) 한국과학기술원 산업및시스템공학과 명예교수
- 관심분야: Product Lifecycle Management, Text Mining, Ontology/Knowledge Based System