applied sciences

MDPI

Article

A Named Entity and Relationship Extraction Method from Trouble-Shooting Documents in Korean

Minkyu Jeong [1], Hyowon Suh [1], Heejung Lee [2] and Jae Hyun Lee [3,*]

[1] Department of Industrial and System Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
[2] Division of Interdisciplinary Industrial Studies, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Republic of Korea
[3] Department of Industrial Engineering, Daegu University, 201 Daegudae-ro, Jinryang-eup, Kyongsan-si 38453, Republic of Korea
* Correspondence: jaehyun.lee@daegu.ac.kr; Tel.: +82-53-850-6544

Abstract: In enterprises operating large-scale equipment, such as plants, maintenance workers must quickly and accurately find and understand the information in the equipment maintenance documents to perform maintenance tasks effectively. If the equipment maintenance documents include sentences with semantically ambiguous expressions, it will interfere with the maintenance knowledge search, and it may affect the maintenance performance of engineers. In order to solve these problems, text-based research of maintenance documents have been done to extract the key information or knowledge from these documents. Previous studies focused on finding the technical terminologies or calculating the similarity of documents using named entity recognition approaches. This paper proposes a method to extract knowledge of not only the technical terminologies but also their relations. The proposed method uses a rule-based approach that can be applied to the results of a named entity recognition approach and a dependency parsing approach. The named entity recognition approach found technical terms and the dependency parsing approach provided sentence structure information, so that the proposed method showed that a set of rules can extract maintenance knowledge, including entities and their relations. Trouble-shooting documents in the field were used as an experiment to demonstrate the effectiveness of the proposed method, and the experiment showed the possibility of practical use of the proposed method.

Keywords: dependency parsing; equipment maintenance documents; named entity recognition

Citation: Jeong, M.; Suh, H.; Lee, H.; Lee, J.H. A Named Entity and Relationship Extraction Method from Trouble-Shooting Documents in Korean. *Appl. Sci.* **2022**, *12*, 11971. https://doi.org/10.3390/app122311971

Academic Editor: Christos Bouras

Received: 7 October 2022
Accepted: 21 November 2022
Published: 23 November 2022

1. Introduction

Maintenance operators in a plant need to quickly search equipment documents and information for accurate and efficient facility maintenance. Correct information and documentation about the equipment can save time and effort for maintenance operators. If a defect occurs in a piece of equipment, it may waste substantial production time and resources. Maintenance operators can prevent equipment failure or reduce equipment downtime by performing correct maintenance activities in a timely manner.

Many plants archive maintenance documents in a file system, and maintenance documents are managed in units of files. Maintenance information about a piece of equipment may exist in separate files, or one file may have maintenance information on several pieces of equipment. Separated and duplicated maintenance data in file systems may hinder maintenance operators from effectively performing a maintenance task. If a part of a specific piece of equipment has a defect and it is necessary to search for the cause and solution, an operator must first find the corresponding maintenance documents, and then sequentially search the content in the documents until the information of interest is found. It could be useful for a maintenance operator if the maintenance data for a specific piece

of equipment could be found by automatically searching texts in documents with natural language processing.

Some prior efforts have been made to extract maintenance knowledge from documents in order to efficiently manage equipment-related documents. A framework [1] to extract failure knowledge from failure reports has been proposed. The framework extracts failure knowledge by calculating the similarity of failure sentences, which are represented in normalized expressions. There were also two studies to extract maintenance knowledge from text documents in the engineering domain. The first of these studies [2] extracted ship failure knowledge from unstructured text data by using a named entity recognition (NER) approach. The second study [3] also implemented an NER approach to extract failure knowledge from maintenance logs of an air compressor.

Knowledge extraction from documents involves an understanding of the semantic relations between items. Relation extraction from text plays a crucial role in natural language processing applications, including engineering-domain knowledge discovery. Traditionally, extracting the relations among entities in documents has been investigated as two main tasks: NER and relation extraction. This approach involves building two distinct models for entity recognition and relation extraction, and then optimizing them jointly by sharing key parameters [4–7]. According to previous research, the performance of entity recognition is generally good, but relation extraction tasks remain challenging. The dependency parsing approach examines word dependencies to discover a sentence's grammatical structure, and it delivers substantial structural information that has been shown to be useful for extracting relations among entities in documents. Many dependency parsing models have been developed and implemented in various sectors, but they have not been applied to engineering documents. Developing a dependency parser for maintenance documents in a short period of time is hard, and it is also not easy to modify and utilize existing dependency parsers for maintenance documents.

Existing approaches [1–3] to extract knowledge from maintenance documents have focused on finding sentences by calculating the similarity of sentences or recognized named entities. They did not consider the relationships among the named entities in a sentence. The relationship among the named entities could be useful to precisely find parts, failures, or repair actions. Therefore, this paper proposes an approach that adopts both the NER and dependency parsing approaches, joining them with simple rules. The proposed approach showed that the extracted knowledge from maintenance documents could represent not only the meaningful entities (parts, failures, repair actions) but also their relations, such as is-part-of, is-failure-of, and is-action-of. The proposed approach was implemented using Korean Language Understanding Evaluation (KLUE) [8], a collection of eight Korean natural language understanding tasks, including NER, dependency parsing, and others.

This paper is organized as follows. Previous research is explored in Section 2. Section 3 presents an overview of the proposed approach and how the pre-processed data were prepared. Sections 4–6 describe each step in the proposed approach with examples: Section 4 explains how to apply the NER approach to maintenance documents; Section 5 shows how the relationships among entities were captured by dependency parsing methods; and Section 6 proposes a rule-based approach to infer maintenance knowledge from the processed data. Section 7 shows the results of the experiment, and the conclusions and future studies are presented in Section 8.

2. Previous Research

2.1. Named Entity Recognition

NER is a natural language processing method to identify conceptual objects from terms in text. Conditional Random Field (CRF), Bidirectional Long- and Short-Term Memory (BiLSTM), or Bidirectional Encoder Representations from Transformers (BERT) models could be used to implement the NER approach. NER refers to a technique used in natural language processing that recognizes phrases containing specific meanings within sentences or documents [9]. Most information extraction applications begin with the identification

and categorization of named things inside a text. Naming an entity makes it possible to refer to everything by its proper name. This technique of named entity recognition involves identifying and categorizing entities based on their category.

According to the definition of NER recorded in the Telecommunications Dictionary maintained by the Telecommunications Technology Association [10], NER is the technique of recognizing, extracting, and sorting phrases (named entities) referring to pre-defined people, companies, locations, time, and units from documents. In this paper, the term "NER" will denote the process of defining equipment components, failure status, and solutions to failure status as entities and extracting them from documents.

Multiple methods [11,12] exist for conducting NER, including but not limited to rule-based methods that utilize rules pre-defined by the user, unsupervised learning methods that apply statistical methods to the input text data without any pre-defined rules or entity names, and feature-based supervised learning that takes in supervised data with labels and applies them to classification or sequence labeling problems.

This paper utilized feature-based supervised learning to conduct NER. CRF [13] and the Korean BERT-based model KLUE [8] were used to extract equipment, equipment components, failure status, and solutions to failure statuses from equipment maintenance documents.

2.2. Dependency Parsing

Dependency parsing is a method of analyzing grammatical composition and/or phrases by identifying dependencies between words within a sentence [14]. Dependency parsing is conducted by comparing two words within a phrase, mapping the modifier and the modified word with a dependency relationship, and specifying the dependency relationship type to clarify the syntactic relationship between two words. The word receiving the modification is referred to as the head or the governor, while the word that modifies the head is referred to as the dependent or the modifier. There are also words that do not modify any other word and are treated as modifying a virtual root node.

After dependency parsing, every word becomes a dependent of at least one node, with no cyclic relationships. Thanks to the usage of these dependency relationships, dependency parsing demonstrates an ability to resolve ambiguities caused by various interpretations of which words are modified by another word in a sentence; therefore, it is widely used in natural language processing research.

The two main ways of conducting recognition are transition-based dependency parsing and graph-based dependency parsing. Transition-based dependency parsing procedurally generates a dependency parsing tree by analyzing the dependency relationships in order [15]. The way of making a dependency parsing tree consists of shift calculation, followed by LEFT-ARC and RIGHT-ARC calculation. The shift calculation moves each word within a phrase from the buffer to the stack, and LEFT-ARC and RIGHT-ARC calculation determines the relationship between the words that have been moved. Classical transition-based dependency parsing is unable to take into consideration the meaning of a word or its part-of-speech (POS) tag. However, researchers have actively sought to improve these limitations, such as utilizing POS tag information and ARC-labels as features in the neural network to improve accuracy and runtime [16]. Graph-based dependency parsing considers the dependency relationships between all possible word pairs within a sentence to create all possible dependency trees of the sentence and compute the parsing result using the most probable dependency tree [17]. Since this method considers all possible cases, its runtime is slower than that of transition-based dependency parsing.

Some previous studies have used Korean dependence syntax analysis. Kwak et al. [18] proposed a dependency parsing approach to infer "subject-predicate-object" triples from texts in Korean. Kim et al. [19] also proposed a rule-based approach with a dependency parsing method to extract the necessary phrases from texts in Korean. Both studies showed that dependency syntax analysis can infer relationships among words, but they focused on a syntactic analysis based on the structural relationships in each sentence rather than

extracting specific entities. This paper used the dependency parser in KLUE [8], which is a transition-based dependency parsing method.

2.3. Text-Based Research for Engineering Documents

Baek et al. [20] reviewed several text-based studies in the construction domain. According to their classification of previous works, entity recognition was used to find the technical terminology, and then a domain ontology was developed by domain experts to solve semantic heterogeneity in the technical terms [21]. Since ontology development requires substantial time and effort, natural language processing models were used to recognize the technical terms [22,23]. Moon et al. [24] developed a construction specification review system that used an NER model with a CRF layer to match the technical terms in different specification documents.

As an example of the NER approach for maintenance documents, Jie and Lu [2] applied NER to Chinese ship fault data, and the proposed method could accurately find named entities. Chen et al. [3] developed a benchmark dataset of air compressor fault diagnoses for knowledge extraction. The dataset was collected from air compressor maintenance log sheets. NER approaches were tested with the proposed dataset, and they showed potential for compressor fault diagnosis.

Previous text-based research for engineering documents showed that the NER approach can be useful to find technology terms as domain entities. The NER approach could resolve the semantic heterogeneity of single terms, but it does not resolve the semantic ambiguity of the relations between domain entities in a sentence. Therefore, this paper proposes a rule-based approach that combines the NER and dependency parsing methods.

3. Overview of the Proposed Approach

3.1. Maintenance Knowledge Extraction

Figure 1 shows the proposed approach for extracting maintenance knowledge from equipment maintenance documents. The first step is to pre-process the equipment maintenance documents, so that texts about failure modes and repair actions are selected from troubleshooting descriptions in the maintenance documents. The selected text becomes training data with which the NER model of KLUE (KLUE-NER) is trained. The trained NER model classifies terms into equipment, part, failure, and action entities. The dependency parser of KLUE (KLUE-DP) is applied to the selected text to analyze the syntactic relationships among terms. The syntactic relationships are used to infer the semantic relations among the named entities. The semantic relations among the named entities are inferred by the proposed rules. The extracted relations among the named entities are provided to an engineer as maintenance knowledge.

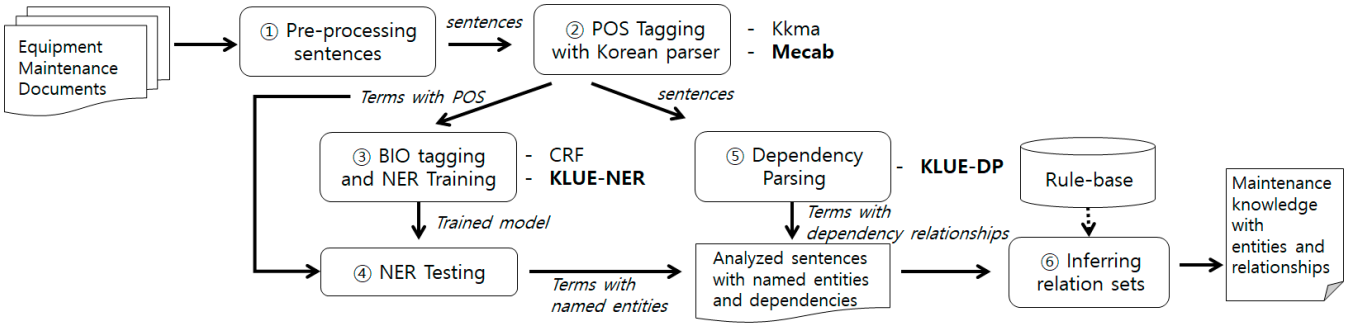


Figure 1. The proposed approach for relation set extraction.

3.2. Pre-Processing Data

The data used in this paper were acquired from the actual documents used in domestic plant enterprises for maintenance tasks. A separate maintenance document was used for each piece of equipment, and in the document, the failure modes that may occur with the

equipment and the corresponding repair actions are described in the form of a table in the item called "troubleshooting". The failure modes and the repair actions are written in a semi-structured manner as words or a list of sentences within the tables, with different special characters or numbers for each document.

As a pre-processing step, the troubleshooting sections in each document, which were separated by types of equipment, were merged into one. The data for analysis were extracted by sentences from the merged troubleshooting section. Special characters and unnecessary additional explanations were removed from the sentences, and grammatical issues such as spacing and typos were corrected manually using a Korean grammar checker. Synonyms in English, used in parentheses for further explanation, could lower the performance of the NER and were hence removed.

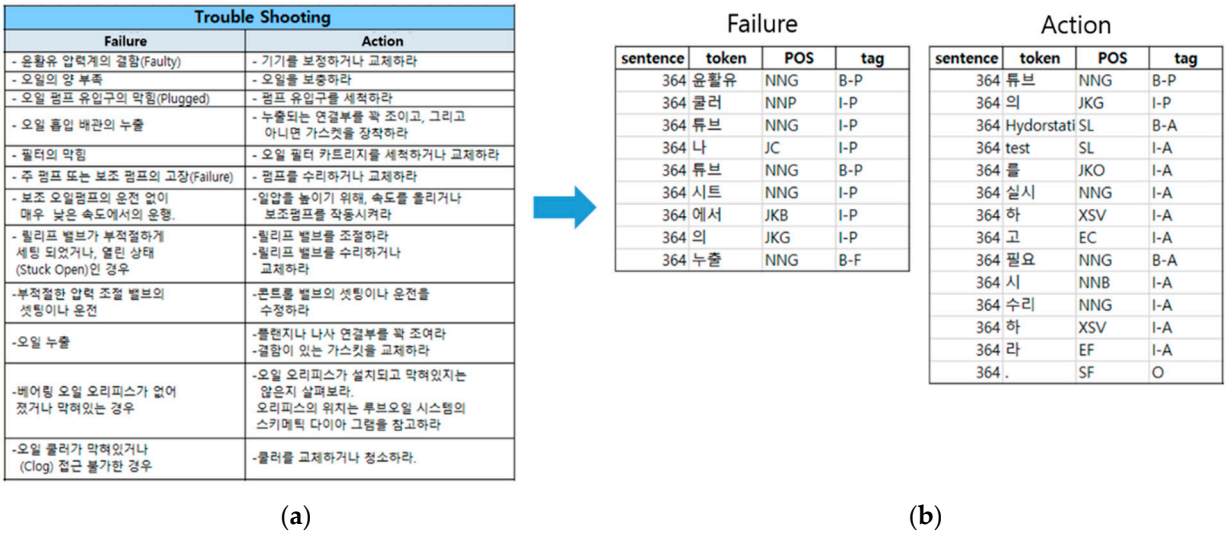
After the correction process, each sentence was tokenized using the open-source Kkma and Mecab morpheme analyzers. The morphemes and their POS information acquired by the analyzers were then utilized in the subsequent step.

4. Named Entity Recognition

The named entities that were to be detected for equipment maintenance knowledge extraction from the generated tokens were defined as follows.

1. PART: Equipment itself, or the components of the equipment (e.g., pressure gauge, valve, turbine, nozzle).
2. FAILURE: Equipment malfunction or failure mode (e.g., clogging, loosening, damaged, coarse surface).
3. ACTION: Actions that must be taken to resolve a failure mode (e.g., control, replace, disassemble).
4. (ETC): Words that refer to a specific object, but are not included in PART, FAILURE, or ACTION (e.g., supplier, pressure, air).
5. O: Words that are not included in PART, FAILURE, ACTION, or (ETC) (e.g., and, therefore).

In order to extract the entities as defined in this section, the tokens from the pre-processed data were tagged using the BIO method. The BIO method attaches "B-named entity" to the beginning, "I-entity name" to the inside of entities, and "O" for non-entity tokens. A <BOS> ("beginning of sentence") tag was also attached in addition to the BIO tag to indicate the start of a sentence. Figure 2 shows a training data sample from an equipment maintenance document in Korean that was pre-processed and BIO-tagged.



Trouble Shooting	
Failure	**Action**
- 윤활유 압력계의 결함(Faulty)	- 기기를 보정하거나 교체하라
- 오일의 양 부족	- 오일을 보충하라
- 오일 펌프 유입구의 막힘(Plugged)	- 펌프 유입구를 세척하라
- 오일 흡입 배관의 누출	- 누출되는 연결부를 꽉 조이고, 그리고 아니면 가스킷을 장착하라
- 필터의 막힘	- 오일 필터 카트리지를 세척하거나 교체하라
- 주 펌프 또는 보조 펌프의 고장(Failure)	- 펌프를 수리하거나 교체하라
- 보조 오일펌프의 운전 없이 매우 낮은 속도에서의 운행.	-일압을 높이기 위해, 속도를 올리거나 보조펌프를 작동시켜라
- 릴리프 밸브가 부적절하게 세팅 되었거나, 열린 상태 (Stuck Open)인 경우	-릴리프 밸브를 조절하라 -릴리프 밸브를 수리하거나 교체하라
-부적절한 압력 조절 밸브의 셋팅이나 운전	-콘트롤 밸브의 셋팅이나 운전을 수정하라
-오일 누출	-플랜지나 나사 연결부를 꽉 조여라 -결함이 있는 가스킷을 교체하라
-베어링 오일 오리피스가 없어 졌거나 막혀있는 경우	-오일 오리피스가 설치되고 막혀있지는 않은지 살펴보라. 오리피스의 위치는 루브오일 시스템의 스키메틱 다이아 그램을 참고하라
-오일 쿨러가 막혀있거나 (Clog) 접근 불가한 경우	-쿨러를 교체하거나 청소하라.

(**a**)

Failure

sentence	token	POS	tag
364	윤활유	NNG	B-P
364	쿨러	NNP	I-P
364	튜브	NNG	I-P
364	나	JC	I-P
364	튜브	NNG	B-P
364	시트	NNG	I-P
364	에서	JKB	I-P
364	의	JKG	I-P
364	누출	NNG	B-F

Action

sentence	token	POS	tag
364	튜브	NNG	B-P
364	의	JKG	I-P
364	Hydorstati	SL	B-A
364	test	SL	I-A
364	를	JKO	I-A
364	실시	NNG	I-A
364	하	XSV	I-A
364	고	EC	I-A
364	필요	NNG	B-A
364	시	NNB	I-A
364	수리	NNG	I-A
364	하	XSV	I-A
364	라	EF	I-A
364	.	SF	O

(**b**)

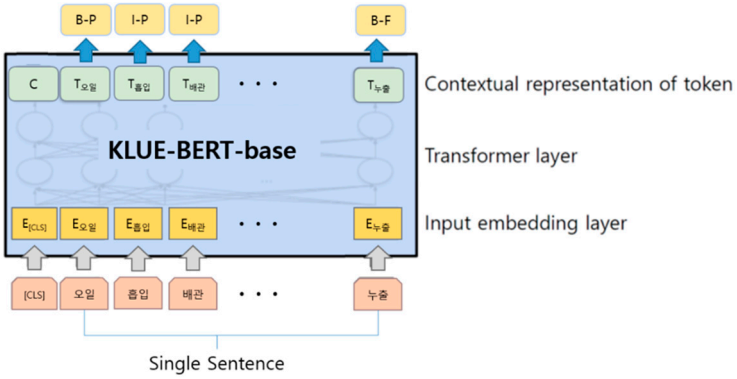
Figure 2. Entity name tagging example: (**a**) original equipment maintenance document example in Korean; (**b**) input examples for the NER model.

The CRF model involves selecting the BIO tag with the maximum likelihood $P(y|x)$, formally defined in Equation (1).

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_t \sum_k \lambda_k f_k(y_{t-1}, y_t, x)\right) \quad (1)$$

In this equation, λ_k represents the weight given to each feature during the learning phase; $f_k(y_{t-1}, y_t, x)$ is a feature function whose value could be 0 or 1; $Z(x)$ is a normalization factor; x denotes the sequence of words in a sentence of documents; and y denotes the BIO tag sequence.

This paper used KLUE [8], which is a BERT model trained on Korean data that has demonstrated remarkable performance in natural language processing. The network takes tokens and BIO tagging values as sequential inputs, while training utilizes pretrained embedding values. The output layer contains a classifier that outputs a tag value for each token. Figure 3a shows a schematic of the model used in this paper, and Figure 3b shows a sample of a successful NER using the model trained on the test dataset.



(**a**)

Failure

```
[[('오일', 'B-P'),
  ('흡입', 'I-P'),
  ('배관의', 'I-P'),
  ('누출', 'B-F')],
 [('느슨해지거나', 'B-F'),
  ('파손된', 'B-F'),
  ('볼트', 'B-P')]]
```

Action

```
[[('릴리프', 'B-P'),
  ('밸브를', 'I-P'),
  ('수리하거나', 'B-A'),
  ('교체하라', 'B-A')],
 [('축이나', 'B-P'),
  ('축', 'B-P'),
  ('슬리브를', 'I-P'),
  ('세척하고', 'B-A'),
  ('다듬어라', 'B-A')]]
```

(**b**)

Figure 3. (**a**) NER model structure; (**b**) example of an NER result.

5. Dependency Parsing

When working on equipment maintenance or searching for information in maintenance documents, the operator must know the semantic relationships among the entities in the domain to clearly understand the text. For automatic processing, it is necessary to accurately understand the generative syntax of the sentences or words composed of overlapping structures, so that the situation at hand can effectively be understood and appropriate countermeasures can be taken. To this end, this paper uses dependency parsing to identify dependency relations among the entities in a sentence that indicates the failure mode and the repair action.

This study used the dependency parser of KLUE (KLUE-DP) for dependency parsing; KLUE-DP is a system built by annotating a total of 10,000 sentences, including news and review data, that conducts dependency parsing using the Korean syntactic tag sets provided by the Telecommunications Technology Association. The transition-based dependency parsing technique [8] used in this paper has a pre-trained Korean model as a baseline to generate word vectors, and it uses biaffine attention to predict heads and bilinear attention to predict modifiers.

The dependency relation tag set for Korean consists of nine syntactic tags as follows: "L: left sign (left bracket character and left open quote)"; "R: right sign (right bracket character and right close quote)"; "NP: nominal (noun, pronoun, numeral)"; "VP: predicate (verb, adjective, auxiliary predicate)"; "AP: adverbial phrase"; "VNP: noun+ 이다/positive copula"; "DP: determiner phrase"; "IP: interjection phrase"; "X: pseudo phrase or symbol". There are also six functional tags: "SBJ (subject)"; "OBJ (object)"; "MOD (modifier)"; "AJT

(adverb)"; "CMP (complement)"; "CNJ (conjunctive)" [25]. Possible dependency relations in Korean sentences are shown in Table 1.

Table 1. Korean dependency relation tag set.

Notation	Dependency Relationship	Notation	Dependency Relationship
L	Left sign	AP_OBJ	Adverbial_Object
R	Right sign	AP_MOD	Adverbial phrase_Modifier
NP_SBJ	Nominal_Subject	AP_CMP	Adverbial phrase_Complement
NP_OBJ	Nominal_Object	VNP_SBJ	Positive copula_Subject
NP_MOD	Nominal_Modifier	VNP_OBJ	Positive copula_Object
NP_AJT	Nominal_Adverb	VNP_MOD	Positive copula_Modifier
NP_CMP	Nominal_Complement	VNP_AJT	Positive copula_Adverb
NP_CNJ	Nominal_Conjunctive	VNP_CMP	Positive copula_Complement
VP_SBJ	Predicate_Subject	VNP_CNJ	Positive copula_Conjunctive
VP_OBJ	Predicate_Object	DP_SBJ	Determiner phrase_Subject
VP_MOD	Predicate_Modifier	DP_OBJ	Determiner phrase_Object
VP_AJT	Predicate_Adverb	DP_MOD	Determiner phrase_Modifier
VP_CMP	Predicate_Complement	DP_AJT	Determiner phrase_Adverb
VP_CNJ	Predicate_Conjunctive	DP_CMP	Determiner phrase_Complement
AP_SBJ	Adverbial phrase_Subject	DP_CNJ	Determiner phrase_Conjunctive

6. Rule-Based Approach for Knowledge Extraction

6.1. Relation Sets and Rules to Find the Relations

Maintenance documents have multiple sentences that express maintenance knowledge, which can be expressed with simplified phrases if key entities are extracted from the sentences. Table 2 shows the simplified formats of maintenance knowledge, which consists of key entities such as "PART", "FAILURE", and "ACTION". The simplified formats are sets of relations between entities, so they are called "relation sets" in this paper. The "PART-FAILURE" and "FAILURE-PART" relation sets are simplified phrases of the sentences in the failure mode section of maintenance documents. The "PART-FAILURE", "FAILURE-ACTION", and "FAILURE-PART-ACTION" relation sets are simplified phrases of the sentences in the repair action section of maintenance documents.

Table 2. Pre-defined relation sets describing failure modes and repair action.

ID	Relation Set	Description	Content Type
1	PART-FAILURE	Part has a failure	Failure Mode
2	FAILURE-PART	A failed part	
3	PART-ACTION	Repair a part	Repair Action
4	FAILURE-ACTION	Fix a failure	
5	FAILURE-PART-ACTION	Repair a failed part	

The key entities of relation sets could be found by applying NER methods. The relation between entities should be inferred from the syntactical structure of a sentence, which could be provided by the results of dependency parsing. Rules using the syntactical structure can be pre-defined to find the relations between entities. Five rules were defined for the five relation sets. Rules are explained with variables and functions. "$words_i$" is a variable for a set of words that are recognized as "PART", "FAILURE", or "ACTION". "*DP*" is a mapping function of two words for their dependency relationship notations.

1. Rule #1: Given that $words_a \in$ PART, $words_b \in$ FAILURE in a sentence,

IF $DP(words_a, words_b)$ = 'NP_MOD' or 'NP_SBJ' or 'NP_AJT',
THEN $(words_a, words_b) \in$PART-FAILURE

Rule #1 is for recognizing the relations of the failure mode data, such as "PART 의[JKG] FAILURE", PART 이/ 가[JKS] FAILURE", or "PART 에[JKB] FAILURE", as being part of the PART-FAILURE set regardless of differences in their forms. For example, as shown in Table 3, PART, as in "튜브- 시트에서의 (tube/NNG − sheet/NNG + JKB + JKG)", and FAILURE, as in " 누출 (leaks/NNG)", are in the NP_MOD dependency relation, and their relation is included in the PART-FAILURE set.

Table 3. An example of the results by applying Rule #1.

ID	1	2	3
Terms	튜브	시트에서의	누출
BIO Tag	B-PART	I-PART	B-FAILURE
DP(i,j)	DP(1,2) = 'NP'	DP(2,3) = 'NP-MOD'	DP(3,3) = 'NP'
Word Entity	PART		FAILURE
Relation Set	(튜브 시트에서의, 누출) ∈ PART-FAILURE		

2. Rule #2: Given that $words_a \in$ FAILURE and $words_b \in$ PART in a sentence,

IF $DP(words_a, words_b)$ = 'VP_MOD',
THEN $(words_a, words_b) \in$ FAILURE-PART

Rule #2 is for recognizing the relations of the failure mode data such as "FAILURE 한[JKG] PART". For example, as shown in Table 4, FAILURE, as in " 파손된 (damaged/VP_MOD), and PART, as in " 볼트 (bolt/NP_SBJ)", are in the VP_MOD dependency relationship, and their relation is included in the FAILURE-PART set.

Table 4. An example of the results by applying Rule #2.

ID	1	2	3
Terms	느슨해지거나	파손된	볼트
BIO Tag	B-FAILURE	B-FAILURE	B-PART
DP(i,j)	DP(1,2) = 'VP'	DP(2,3) = 'VP-MOD'	DP(3,3) = 'NP'
Word Entity	FAILURE	FAILURE	PART
Relation Set	(파손된, 볼트) ∈ FAILURE-PART		

3. Rule #3: Given that $words_a \in$ PART and $words_b \in$ ACTION in a sentence,

IF $DP(words_a, words_b)$ = 'NP_OBJ' or 'NP_SBJ',
THEN $(words_a, words_b) \in$ PART-ACTION

Rule #3 is for recognizing the relations of the repair action data such as "PART 을/ 를[JKO] ACTION 하라(한다)[VP]."or "PART 은/는/이/ 가[JKS] ACTION 되어야 한다[VP]." For example, as shown in Table 5, PART, as in " 축을 (axes/NP_OBJ", and ACTION, as in " 곧게 (straighten/VP_AJT) 수정하거나 (repair/VP)", are in the NP_OBJ dependency relationship, and their relation is included in the PART-ACTION set.

Table 5. An example of the results of applying Rule #3.

ID	1	2	3	4
Terms	축을	곧게	수정하거나	교체하라
BIO Tag	B-PART	B-ACTION	I-ACTION	B-ACTION
DP(i,j)	DP(1,3) = 'NP_OBJ'	DP(2,3) = 'VP_AJT'	DP(3,4) = 'VP'	DP(4,4) = 'VP'
Word Entity	PART	ACTION		ACTION
Relation Set	(축을, 곧게 수정하거나) ∈ PART-ACTION			

4. Rule #4: Given that $words_a \in$ FAILURE and $words_b \in$ ACTION in a sentence,

IF $DP(words_a, words_b)$ = 'NP_OBJ' or 'VP_OBJ',
THEN $(words_a, words_b) \in$ FAILURE-ACTION

Rule #4 is for recognizing the relations of the repair action data such as "FAILURE 을/ 를[JKO] ACTION 하라(한다)[VP]." For example, as shown in Table 6, FAILURE, as in " 모든 (all/NP_MOD), 응축된 (condensed/VP_MOD), and 액체들을 (liquids/NP_OBJ)", and ACTION, as in " 드레인 (drain/VP_MOD) 하여라 (do/VP)", are in the NP_OBJ dependency relationship, and their relation is included in the FAILURE-ACTION set.

Table 6. An example of the results by applying Rule #4.

ID	1	2	3	4	5
Terms	모든	응축된	액체들을	드레인	하여라
BIO Tag	B-FAILURE	I-FAILURE	I-FAILURE	B-ACTION	I-ACTION
DP(i,j)	DP(1,3) = 'DP'	DP(2,3) = 'VP_MOD'	DP(3,5) = 'NP_OBJ'	DP(4,5) = 'NP_OBJ'	DP(5,5) = 'VP'
Word Entity	FAILURE			ACTION	
Relation Set	(모든 응축된 액체들을, 드레인 하여라) ∈ FAILURE-ACTION				

5. Rule #5: Given that $words_a \in$FAILURE, $words_b \in$ PART, $words_c \in$ ACTION in a sentence,

IF $DP(words_a, words_b)$ = 'VP_MOD' and
$DP(words_b, words_c)$ = 'NP_OBJ' or 'NP_SBJ',
THEN $(words_a, words_b, words_c) \in$ FAILURE-PART-ACTION

Rule #5 is for recognizing the relations of the repair action data such as "FAILURE 한[JKG] PART 을/ 를[JKO] ACTION 하라(한다)[VP]." For example, as shown in Table 7, FAILURE, as in " 느슨한 (loosen/VP_MOD)", PART, as in " 부품들을 (parts/NP_OBJ)", and ACTION, as in " 수리하거나" (repair/VP)", between FAILURE-PART are entities in the VP_MOD dependency relationship, as well as between PART and ACTION. The entities are in the NP_OBJ dependency relationship, and their relation is included in the FAILURE-PART-ACTION set.

Table 7. An example of the results by applying Rule #5.

ID	1	2	3	4
Terms	느슨한	부품들을	수리하거나	교체하라
BIO Tag	B-FAILURE	B-PART	B-ACTION	B-ACTION
DP(i,j)	DP(1,2) = 'VP_MOD'	DP(2,3) = 'NP_OBJ'	DP(3,4) = 'VP'	DP(4,4) = 'VP'
Entity	FAILURE	PART	ACTION	ACTION
Relation Set	(느슨한, 부품들을, 수리하거나) ∈ FAILURE-PART-ACTION			

6.2. Rules to Find Additional Relations for Extended Phrases

When building a relation set by applying the rules defined in Section 6.1., relations can be found accurately for short sentences. However, if an expression is semantically related to two or more expressions, there is a limitation that its relation cannot be inferred directly. For the example in Table 5, not only a PART-ACTION relation '(축을, 곧게 수정하거나)' should be found, but also a PART-ACTION relation '(축을, 교체하라)' should be found because the ending of the word '~ 거나' denotes the OR relationship.

Additional rules in this sub-section are defined in order to infer relations that are expressed as extended phrases in a sentence. The extended phrases in a sentence generally use specific POS such as 'JC', which refers to an ending word for a connection in Korean. The additional rules can be defined by using a *POS* function, which maps from words to a set of POSs for the terms in the words.

6. Rule #6: Given that $words_a \in$ PART AND

$(words_b, words_c) \in$PART-FAILURE in a sentence,
IF 'JC' $\in POS(words_a)$ OR 'NP_CNJ' $\in DP(words_a, words_b)$,
THEN $(words_a, words_c) \in$ PART-FAILURE

Before applying Rule #6, " 누출 (leaks/NNG)" is only connected to " 튜브 (tube/NNG) 시트에서의 (sheet/NNG_MOD)". When Rule #6 is applied, as shown in Table 8, similar to "(sheet/NNG_MOD)", " 윤활유 (lubricant/NNG) 쿨러 (cooler/NNP) and 튜브나 (tube/NNG + JC)" can also be identified as PART, and they both have the NP_CNJ relation, which enables them to be considered as extended phrases. Thus, " 누출 (leaks/NNG)", which has established the PART-FAILURE relation set only with " 튜브- 시트에서의 (tube/NNG-sheet/NNG + JKB + JKG)", can also form a PART-FAILURE relation set with " 윤활유 (lubricant/NNG 쿨러 (cooler/NNP) 튜브나 (tube/NNG + JC)" at the same time.

Table 8. An example of the results by applying Rule #6.

ID	1	2	3	4	5
Terms	쿨러	튜브나	튜브	시트에서의	누출
POS	NNP	NNG + JC	NNG	NNG + JKB + JKG	NNG
BIO Tag	B-PART	I-PART	B-PART	I-PART	B-FAILURE
DP(i,j)	DP(1,2) = 'NP'	DP(2,4) = 'NP_CNJ'	DP(3,4) = 'NP'	DP(4,5) = 'NP-MOD'	DP(5,5) = 'NP'
Word Entity	PART		PART		FAILURE
Relation Set	(쿨러 튜브에서의, 누출) ∈ PART-FAILURE				

Rules #7 to #10 are also defined in order to infer relations from extended phrases, such as Rule #6.

7. Rule #7: Given that $words_a \in$ FAILURE AND

   ($words_b$, $words_c$) ∈ FAILURE-PART in a sentence,
   IF 'JC' ∈ $POS(words_a)$ OR 'VP_CNJ' ∈ $DP(words_a, words_b)$,
   THEN ($words_a$, $words_c$) ∈ FAILURE-PART

8. Rule #8: Given that $words_a \in$ PART AND

   ($words_b$, $words_c$) ∈ PART-ACTION in a sentence,
   IF 'JC' ∈ $POS(words_a)$ OR 'NP_CNJ' ∈ $DP(words_a, words_b)$,
   THEN ($words_a$, $words_c$) ∈ PART-ACTION

9. Rule #9: Given that $words_a \in$ FAILURE AND

   ($words_b$, $words_c$) ∈ FAILURE-ACTION in a sentence,
   IF 'JC' ∈ $POS(words_a)$ OR 'VP_CNJ' ∈ $DP(words_a, words_b)$,
   THEN ($words_a$, $words_c$) ∈ FAILURE-ACTION

10. Rule #10: Given that $words_d \in$ ACTION,

    ($words_a$, $words_b$, $words_c$) ∈ FAILURE-PART-ACTION in a sentence,
    IF 'JC' or 'EC' ∈ $POS(words_c)$ and $DP(words_b, words_c)$ = 'VP' or 'VP_CNJ',
    THEN ($words_a$, $words_b$, $words_d$) ∈ FAILURE-PART-ACTION

6.3. Implementation of the Rule-Based System

A rule-based system was implemented for the text analysis with the proposed rules. The system was implemented with Python 3.7 to use the existing KLUE modules, and the user interface was implemented with C# programming on .NET framework 4.8.

The rule-based system can validate the application of the proposed rules to maintenance documents. The system receives text inputs that have sentences about failure modes and repair actions. Each sentence is pre-processed for text analysis, and then NER and DP processing is conducted to give the user results (data) in the form of a spreadsheet. The rules are applied to the resulting data, and then each type of relation set is shown in the corresponding tabs (Figure 4). The implemented system was used for an experiment to show the validity of the proposed method with practically used text data for troubleshooting at a plant company. The next section explains the experiment's setting and results.

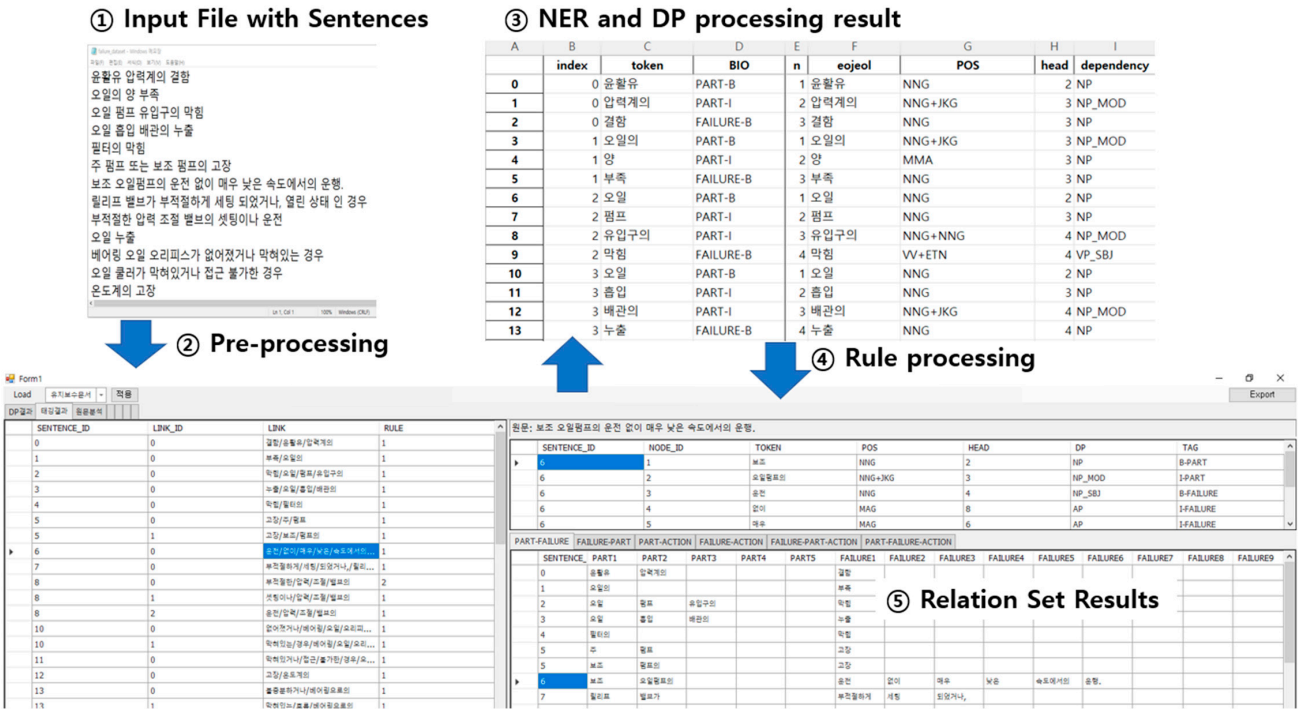


Figure 4. Screenshots of the proposed rule-based system.

7. Experiment and Results

This paper analyzed equipment maintenance documents used in domestic plant enterprises. In total, 10 out of 17 equipment documents were selected, excluding duplications. Troubleshooting data were divided into "failure mode" and "repair action", and each data set was organized into sentence units. The data on failure mode consisted of 451 sentences, and the data on repair actions consisted of 449 sentences

In the experiment, combinations of two morpheme analyzers and two NER approaches were compared to find the best match for the troubleshooting sentences in Korean.

Kkma and Mecab as the morpheme analyzer generated different numbers of tokens. For the failure mode sentences, Kkma generated 1956 tokens, while Mecab generated 1774 tokens. For the repair action sentences, Kkma generated 4551 tokens, while Mecab generated 4168 tokens.

When examining the sentences constituting each document, some used a similar vocabulary, but some used many different expressions, making the sentences inconsistent. In addition, several terms were sometimes used in different sentences to represent the same equipment part. In order to determine whether a NER approach can resolve the semantic heterogeneity of terms, the KLUE NER and CRF approaches were trained and tested.

Domain experts provided correct BIO tags for the tokenized sentences. For the failure mode, three types of tags, except for the "ETC" tag, were used: "PART", "FAILURE", and "O". For the repair action, four types were used: "PART", "FAILURE", "ACTION", and "O". In total, 70% of the total sentences were randomly selected to train the KLUE NER model and CRF model. The other 30% of the total sentences were used to test the trained models.

The performance of the combination of the morpheme analyzer and the NER approach was measured by the precision, recall, and F1 score, as defined in the following equations.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

$$F1\ score = \frac{2 \times precision \times recall}{(precision + recall)} \quad (4)$$

True positive denotes the number of correct answers of the trained model for the entities, except "O". *False positive* means the number of wrong answers of the trained model for the entities, except "O". *False negative* means the number of wrong answers of the trained model for the "O"-tagged terms.

Table 9 shows the experimental results. The final F1 score of KLUE-NER with Mecab was 0.8235 for the failure mode data and 0.6888 for the repair action data. The relatively low results for the repair action data may be attributed to the fact that, despite data preprocessing, some unnecessary expressions were still included in the repair action data.

Table 9. A comparison of the combination of NER approaches and morpheme analyzers.

		Kkma		Mecab	
		Failure	**Action**	**Failure**	**Action**
CRF	Precision	0.7440	0.7085	0.7042	0.6746
	Recall	0.7420	0.7218	0.7154	0.6895
	F1 score	0.7430	0.7151	0.7098	0.6820
KLUE-NER	Precision	0.7246	0.6705	0.7712	0.6759
	Recall	0.8315	0.6853	0.8834	0.7021
	F1 score	0.7744	0.6778	0.8235	0.6888

For the recognized entities, relation sets were established by applying the rules that considered the dependency parsing results. Correct relation sets were given by domain experts. The performance results of the knowledge extraction using the proposed rules in Sections 6.1 and 6.2 are presented in Table 10. The number of sentences with accurately recognized entities and relations was 133 out of a total of 179 sentences (74.3%). The proposed method was successful in 56 out of 65 failure mode-related sentences and 77 out of 114 repair action-related sentences, and it obtained a 86.15% and 67.54% accuracy, respectively.

Table 10. Relation set recognition results.

Relation Set	Correct Relation Sets	All Relation Sets	(%)
PART-FAILURE	47	52	90.38
FAILURE-PART	18	18	100
PART-ACTION	68	85	80.00
FAILURE-ACTION	25	26	96.15
FAILURE-PART-ACTION	5	5	100
SUM	163	186	87.63

8. Conclusions

This paper presents an automation method that can extract the following from equipment maintenance documents: (1) parts of the equipment; (2) the failure mode of the equipment; (3) solutions for defect management; and (4) relation sets established based on the semantic relationships between the entities. Using the Korean BERT model, from the failure mode data, "PART" and "FAILURE" were extracted, and from the repair action data, "PART", "FAILURE", and "ACTION" were extracted. Then, relation sets based on dependency relationships were established by applying the proposed rules to the extracted results.

Previous text-based research for engineering documents focused on finding keywords or technical terminologies in sentences. However, knowledge extraction from engineering

documents should consider not only named entities in sentences but also relations between entities. This paper proposes a rule-based approach combining an NER approach and a dependency parsing approach in order to extract both the named entities and their relations. The KLUE NER model and dependency parser were used to implement the proposed approach. The proposed rules were suggested to resolve the semantic heterogeneity of the relations between named entities.

This study has the following limitations, and additional follow-up studies are needed accordingly. In this paper, NER was implemented only for three elements: the equipment itself, or parts of the equipment; the failure mode of the equipment; and its repair action. Information regarding equipment maintenance, however, consists of various elements, and the named entities that can be extracted can also be expanded. In addition, if the expressions represented in the failure mode data and the repair action data differ, even if they refer to the same entity, the repair process would not be performed. This problem can be solved by linking the documents to bills of material or design documents used in the actual plant and manufacturing sectors.

Semantic heterogeneity and inconsistency in technical terminologies and relations could be resolved if a domain ontology was provided. The proposed system could be useful for domain engineers to build a domain ontology gradually and semi-automatically. A study on how to build a domain ontology for text-based research for engineering documents could be conducted in the future based on the proposed approach.

Author Contributions: Conceptualization, H.S. and H.L.; Methodology, H.S., H.L. and J.H.L.; Formal analysis, M.J. and J.H.L.; Writing—original draft, M.J. and J.H.L.; Writing—review & editing, H.L.; Funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the "Developmental project of AI-based gas-oil plant management/maintenance core technology (Project No: 21ATOGC161932-01)" project, funded by the Ministry of Land, Infrastructure & Transport (2022); the "Developmental project of AI-based data analyses of consumer product customer evaluation of consumer product and manufacturing utilization services (Project No: 20009185)"; and the "Developmental project of AI-based digital transformation and extraction technology of engineering design information (Project No: RS-2022-00143813)" funded by the Ministry of Trade, Industry & Energy (2022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, D.; Kang, B.; Lee, S. Failure Knowledge Extraction Framework from Failure Reports in Large Industries. *Asia-Pac. J. Multimed. Serv. Converg. Art Humanit. Sociol.* **2018**, *8*, 955–964. [CrossRef]
2. Jie, Z.; Lu, W. Ship Fault Named Entity Recognition Based on Bilayer Bi-LSTM-CRF. In Proceedings of the 3th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Yantai, China, 16–18 October 2010; pp. 1032–1036.
3. Chen, T.; Zhu, J.; Zeng, Z.; Jia, X. Compressor Fault Diagnosis Knowledge: A Benchmark Dataset for Knowledge Extraction From Maintenance Log Sheets Based on Sequence Labeling. *IEEE Access* **2021**, *9*, 59394–59405. [CrossRef]
4. Miwa, M.; Bansal, M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 1105–1116.
5. Bekoulis, G.; Deleu, J.; Demeester, T.; Develder, C. Adversarial training for multi-context joint entity and relation extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2830–2836.
6. Luan, Y.; He, L.; Ostendorf, M.; Hajishirzi, H. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3219–3232.
7. Lin, Y.; Ji, H.; Huang, F.; Wu, L. A Joint Neural Model for Information Extraction with Global Features. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7999–8009.

8. Park, S.; Moon, J.; Kim, S.; Cho, W.I.; Han, J.; Park, J.; Song, C.; Kim, J.; Song, Y.; Oh, T.; et al. KLUE: Korean Language Understanding Evaluation. In Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems, Online, 6–14 December 2021.
9. Clark, A.; Fox, C.; Lappin, S. *The Handbook of Computational Linguistics and Natural Language Processing*; John Wiley & Sons: Hoboken, NJ, USA, 2012; Volume 118.
10. Telecommunications Technology Association. Information and Communication Terminology. Available online: http://terms.tta.or.kr (accessed on 1 October 2022).
11. Farmakiotou, D.; Karkaletsis, V.; Koutsias, J.; Sigletos, G.; Spyropoulos, C.D.; Stamatopoulos, P. Rule-based named entity recognition for Greek financial texts. In Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries, Kato Achaia, Greece, 22–23 September 2000; pp. 75–78.
12. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [CrossRef]
13. Sutton, C.; McCallum, A. An Introduction to Conditional Random Fields. *Found. Trends Mach. Learn.* **2012**, *4*, 267–373. [CrossRef]
14. Jurafsky, D.; Martin, J.H. *Speech and Language Processing*, 2nd ed.; Prentice Hall: Hoboken, NJ, USA, 2014.
15. Nivre, J. An efficient algorithm for projective dependency parsing. In Proceedings of the Eighth International Conference on Parsing Technologies, Nancy, France, 23–25 April 2003; pp. 149–160.
16. Smith, A.; de Lhoneux, M.; Stymne, S.; Nivre, J. An Investigation of the Interactions Between Pre-Trained Word Embeddings, Character Models and POS Tags in Dependency Parsing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2711–2720.
17. McDonald, R.; Nivre, J. Analyzing and integrating dependency parsers. *Comput. Linguist.* **2011**, *37*, 197–230. [CrossRef]
18. Kwak, S.; Kim, B.; Lee, J.S. Triplet extraction using Korean dependency parsing result. In Proceedings of the Annual Conference on Human and Language Technology, Seoul, Republic of Korea, 7–10 July 2013; pp. 86–89.
19. Kim, H.; Sun, H.; Kim, Y. Development of an Information Extraction System Using the Dependency Analysis. *J. KIISE* **2020**, *47*, 266–275. [CrossRef]
20. Baek, S.; Jung, W.; Han, S.H. A critical review of text-based research in construction: Data source, analysis method, and implications. *Autom. Constr.* **2021**, *132*, 103915. [CrossRef]
21. Lima, C.; Diraby, T.E.; Stephens, J. Ontology-based optimisation of knowledge management in e-Construction. *J. Inf. Technol. Constr.* **2005**, *10*, 305–327.
22. Le, T.; Jeong, H.D. NLP-based approach to semantic classification of heterogeneous transportation asset data terminology. *J. Comput. Civ. Eng.* **2017**, *31*, 1–48. [CrossRef]
23. Nedeljkovic, D.; Kovačević, M. Building a construction project key-phrase network from unstructured text documents. *J. Comput. Civ. Eng.* **2017**, *31*, 1–14. [CrossRef]
24. Moon, S.; Lee, G.; Chi, S. Automated system for construction specification review using natural language processing. *Adv. Eng. Inform.* **2022**, *51*, 101495. [CrossRef]
25. TTAS. *Dependency Tag Sets and Dependency Relation Establishment Methods for Constructing Dependency Tagged Corpora*; TTAK.KO-10.0853; Telecommunications Technology Association: Seonnam, Republic of Korea, 2015.