Rule-based Information Extraction for Mechanical-Electrical-Plumbing-specific Semantic Web

Lang-Tao Wu[1]; Jia-Rui Lin[1]; Shuo Leng[1]; Jiu-Lin Li[2]; Zhen-Zhong Hu[3,1,*]

[1] Department of Civil Engineering, Tsinghua University, Beijing, China, 100084
[2] Beijing Urban Construction Group Company Ltd., Beijing, China, 100088
[3] Shenzhen International Graduate School, Tsinghua University, Shenzhen, China, 518055

Corresponding author: Zhen-Zhong Hu (e-mail: huzhenzhong@tsinghua.edu.cn)

Abstract

Information extraction (IE), which aims to retrieve meaningful information from plain text, has been widely studied in general and professional domains to support downstream applications. However, due to the lack of labeled data and the complexity of professional mechanical, electrical and plumbing (MEP) information, it is challenging to apply current common deep learning IE methods to the MEP domain. To solve this problem, this paper proposes a rule-based approach for MEP IE task, including a "snowball" strategy to collect large-scale MEP corpora, a suffix-based matching algorithm on text segments for named entity recognition (NER), and a dependency-path-based matching algorithm on dependency tree for relationship extraction (RE). 2 ideas called "meta linking" and "path filtering" for RE are proposed as well, to discover the out-of-pattern entities/relationships as many as possible. To verify the feasibility of the proposed approach, 65MB MEP corpora have been collected as input of the proposed approach and an MEP semantic web which consists of 15978 entities and 65110 relationship triples has been established, with an accuracy of 81% to entities and 75% to relationship triples, respectively. A comparison experiment between classical deep learning models and the proposed rule-based approach was carried out, illustrating that the performance of our method is 37% and 49% better than the selected deep learning NER and RE models, respectively, in the aspect of extraction precision.

Keywords: information extraction, MEP, rule match, named entity recognition, relation extraction, natural language understanding, semantic web.

1 Introduction

With the development of the architecture, engineering, construction, and facility management (AEC/FM) industry, a great amount of data has been accumulated in recent years. To break through the traditional analysis and management methods by making use of these data, and thus accelerating

the intelligent development of civil engineering, quite a lot of researches have been conducted to utilize the acquired big data, proving that data-driven engineering knowledge discovery is feasible. However, mining the disorganized data including the large amounts of unstructured data that stored in the form of natural language is of high technical requirements, thus hindering the wide application of raw data. Only if the data is organized to become a knowledge-aware facility for general application can data be utilized by ordinary people.

One typical and important scenario of generating the knowledge is to extract useful information from plain texts expressed in natural language. This information extraction (IE) process is usually divided into the following sub-tasks [1]: named entity recognition (NER), relationship extraction (RE), and event extraction (EE). NER aims to find subjects related to the information from the text, while RE and EE try to link these subjects together to form a complete information group with logics inside. With these tasks, multiple triples can be obtained in the form of {*head entity, relationship, tail entity*} from plain text. These triples can be organized in a graph structure called semantic web [2], or knowledge graph (KG) [3], by further processing. These semantic webs or KGs provide support for some downstream applications, such as information retrieval (IR), question answering (Q&A), etc. They are particularly useful in domains in which there are high requirements for professional knowledge. For example, a triple {total thrust, positive correlation, forward digging speed} indicates that when the forward digging speed is not positively correlated to the total thrust during the operation of a tunnel boring machine, some potential safety incidents might happen because of machine faults or sudden changes in the stratum. This kind of knowledge can be embedded in the control system to improve the safety level of construction.

On the other hand, the mechanical, electrical and plumbing (MEP) engineering, which is a significant component of a building and costing over 65% within the building lifecycle [4], has accumulated a lot of knowledge and experiences in the forms of industry specifications, technical manuals, engineering reports, research literature, etc. These text-based materials are mainly used to describe the relevant characteristics, attributes, working conditions and operation methods of the MEP equipment. Since a lot of MEP documents have been digitalized nowadays, it is possible to apply IE methods to extract MEP information and develop an MEP knowledge system for IR, reasoning or Q&A. For example, the rated operating voltage of a certain device can be extracted from its instruction manual for running condition monitoring.

Nevertheless, there is no existing specific large-scale semantic web for MEP because of lacking sufficiently effective IE approaches due to the following problems. 1) The IE for MEP engineering is a tedious task that requires huge workloads on both raw data collecting and processing. 2) Current IE for common domains mainly relies on deep learning methods, which require large-scale labeled dataset. However, there is no suitable dataset in the MEP field for deep learning or other supervised learning methods. 3) IE studies based on deep learning pay less attention on feature engineering while try to cover all data distributions with one common model. However, the MEP engineering is a professional field in which the information structure and language representation are much more complex than common domains, making it more difficult for IE models to cover all those data distributions.

To solve the above-mentioned problems and provide a novel approach to generate and utilize engineering domain knowledge, this paper proposes a rule-based approach for IE tasks in MEP field, including a “snowball” strategy to collect large scale corpora, a suffix matching strategy for NER and a dependency path matching strategy for RE, so that the MEP domain IE can be processed and

the MEP semantic web can be established efficiently, with sufficient accuracy and considerable data size.

The rest of the paper is organized as follows. A literature review on the IE application for the construction industry and some popular deep learning methods for common domain IE, as well as the rule-based IE in section 2. Afterward, the methodology about the whole technological idea and strategy of this paper is demonstrated thoroughly in section 3. Then, the case study shows the detail about output and some other statistics for every stage in section 4. A comparison experiment between some classical deep learning models and our rule-based approach was also discussed in section 5. Finally, conclusion remarks are presented in Section 6.

2 Literature Review

2.1 IE for the construction industry

IE refers to a text processing technology that identifies specific entities, relationships, events and other information from natural language texts and forms structured output [1]. More specifically, IE takes a string of plain text as input and then outputs some structured data related to the topic of interest to the task. According to the input scale, IE can be further divided into text level extraction, inter-sentence extraction (paragraph level) and single sentence extraction. In addition, IE is usually divided into the following sub tasks. 1) NER, which aims to extract from the text the named entities to represent some specific physical objects such as people, places, institutions, etc., or abstract concepts such as time, price and index, etc. 2) RE, which aims to obtain the relationship between the named entities such as a machine within a system, a property of a machine, subclass, instance, etc. 3) EE is applied to extract multivariate relationships to form a complete event.

These IE tasks are adopted in the construction industry to improve the performance of a building and the efficiency of management. For example, the automatic compliance checking (ACC) aims to extract information from the texts in the design standards, instruction manual and so on [5]. For example, Zhang et al. [6] integrate semantic natural language processing (NLP) and logic reasoning into a unified system for fully automated code checking. One important task within this process is to translate unstructured texts into semi-structured or full structured representations while the specification translation is still considered as the bottlenecks of ACC [7]. Moon et al. conducted the NLP to automated the construction specification review with NER [8] and further recognize the bridge damage from inspection reports using NER based on machine learning algorithms [9]. This kind of document/knowledge retrieval is considered as another hot research interest in the industry [10]. To improve the accuracy of document retrieval, topic mining in advance is proved to be feasible [11][12]. The popular topic mining algorithms include latent dirichlet allocation (LDA) [13] and term frequency-inverse document frequency (TF-IDF) [14].

The semantic web, which is a typical output of IE, has also been widely studied and utilized in the construction industry. Paulheim and Heiko [15] claimed that the combination of BIM and semantic web is an important way to integrate and perfect knowledge in the ACE/FM industry. For example, to support data sharing and exchange, Terkaj and Šojić [16] proposed an approach to rewrite Industrial Foundation Class (IFC) files with ontology web language (OWL), to ensure that

different files can meet the criterion of standard resource description framework (RDF). Similar to this idea, a fuzzy extension of BIM ontologies with ifcRDF and ifcOWL was designed to support imprecise knowledge representation and retrieval [17]. Associated with BIM technology, the semantic web shows greater potential to serve the lifecycle management of buildings. For example, Kim et al. [18] implemented a collaborative information management framework with a semantic web within the building operation and maintenance stage. Ding et al. [19] proposed an ontology-based construction risk knowledge management framework by combining construction safety rules and the BIM environment. Some frameworks were also proposed to integrate the concepts of scan-to-BIM with semantic web to enrich the building information [20], to support specific asset/facility management functions in an integrated and interactive way [21], and to analyze the spatial changes of MEP components automatically based on a relational graph [22]. In the aspect of domain knowledge review and reasoning, Beach et al. [23] implemented a system for experts to define semantic rules and regulatory compliance systems easily. Zhang and EI-Gohary [24] proposed an approach to formalize the representation of regulatory and design information in the form of semantic-based or ontology-based logic clauses to support automated compliance reasoning.

2.2 Common NER/RE

Current popular solutions for NER and RE are mainly based on deep learning. Classical NER solutions rely on sequence labeling models such as the LSTM+CRF [25]. Currently, some optimization algorithms have been proposed to improve this basic model. For example, Chiu and Nichols [26] and Zukov-Gregoric et al. [27] replaced LSTM/RNN with CNN/attention to speed up the training process. Besides, large-scale pre-trained models like BERT [28] or GPT [29] were adopted to encode the input, and to add extra features like lexicon [30][31] and syntactic dependency [32]. There are also researches aiming to improve the efficiency when the sequence labeling model has numerous classes [33][34].

The relation classification (RC) models, such as CNN [35] and other algorithms using BERT and self-attention[36] are generally adopted as a solution for RE. The optimization strategy of these models is similar to the NER while the joint model is an emerging direction to extract entity and relation simultaneously, avoiding the propagation of error between NER and RE. For example, Li and Ji [37] proposed a labeling framework based on sequence labeling and Zheng et al. [38] further optimized this method. The idea of distant supervision [39] is also introduced to RE while it brings a labeling problem to the task itself. Thus, the de-nosing strategies come up, like the multi-instance learning [40]. Lin et al. [36] proposed a selector-classifier framework whose idea is to select correct samples for the classification phase. Some other attempts include modeling selector-classifier as reinforcement learning [41] and generative adversarial learning [42].

As shown in Table 1 and Table 2, we summarize the performance of some classical deep learning solutions as well as their corresponding datasets, and the basic statistical information about some widely used common domain NER/RE datasets. According to Table 1, joint solutions naturally show the worst precision since they must consider NER and RE at the same time. Moreover, the performance of RE is worse than that of NER. The reason might be that RE is more complicated in classification.

Table 1. Representative models of NER/RE with their dataset and performance

Task	Model	Dataset	Classes	Performance
NER	LSTM+CRF [25]	CoNLL-2003	4	F1=90.74%
	Attention+GRU [43]	MSRA	3	F1=94.97%
RE	CNN [35]	Freebase&NYT (part)	53	precision = 78.3%
	Attention+CNN [36]	FreeBase&NYT (part)	53	precision = 72.2%
	RL [41]	NYT(part)	53	F1=42%, precision = 64%
Joint	Pipeline joint [37]	ACE2005	24	precision = 49.5%
	LSTM [38]	NYT	24	F1=52.0%

Note: F1 = 2 * precision * recall ratio / (precision + recall ratio)

Table 2. Some common NER/RE datasets

Task	Dataset	Classes	Scale
NER	MUC conference	9	-
	ACE 2004	24	16000 entities
RE	ACE 2004	24	16771 sentences
	TACRED	42	21784 sentences
	NYT10	53	694491sentences (79% NaN relation)

2.3 Rule-based NER/RE

Existing rule-based NER/RE solutions are mainly based on hand-crafted rules, especially before the popularity of deep learning. Both NER and RE rules are mainly designed upon the syntactic and lexical features, and are considered as efficient, accurate and stable, though these hand-crafted rules design may at the same time of high cost and time consuming [44]. Take NER as an example, Collins and Singer [45] proposed a rule-based NER method and reached 91% precision on the MUC dataset. As for RE, some rule-based researches were proposed from the 1990s to the 2000s and reached precision greater than 80% [46] [47]. However, the efforts spent on these data preparing processes are not feasible for large-scale datasets.

The deep learning methods have been developing rapidly in the past decade, thus resulting in a gradual decrease for studies on rule-based NER/RE. However, deep learning for IE nowadays seems to enter a bottleneck period and the performance improvement gradually slows down. Take NER algorithms in Table 1 as an example, compared to the previous researches in 2016 [25], the latter results in 2020 [43] actually have no significant improvement. Specifically for RE, the performances of latter researches [36][41] are no better than those of classical models [35]. This problem makes it difficult to transfer these deep learning studies from common domains to professional domains where there are few labeled samples [48].

Hence, there are still many rule-based NER/RE methods proposed in recent years, including the rule-based NER [49] and RE [50] for common domains. At the same time, rule-based NER/RE studies have been frequently carried out for the professional fields in recent years. For example, a pre-processed synonym dictionary was proposed to identify protein mentions and potential genes in biomedical texts [51], a dictionary-based approach was proposed for NER in electronic health

records [49] and a rule-based approach was proposed to extract drug information from drug crime news [52]. It seems that these professional rule-based domains NER reach the high-level performance compared to those proposed between 1990-2000.

2.4 Summary of the reviews

The above review demonstrates the importance of applying IE and semantic web technologies to the AEC/FM industry, and a lot of test cases show that knowledge is feasible to level up the intelligence of the industry. Within the process of IE, algorithms for NER and RE are developing in two directions, i.e., based on deep learning models and based on rules, respectively, regardless of common or professional domains. The data-driven knowledge accumulation is popular and providing satisfying supports to common domains. However, there are still some problems remain to be solved to further popularize the knowledge engineering in the construction industry, especially for highly professional domains such as MEP engineering.

It lacks of large-scale dataset for NER and keyword extractions in professional domains. For example, 300 domain-specific sentences were selected in a typical solution of NER for design specification semantic enrichment [53], or 262 inspection records and 287 specification clauses for NER tagging in another study [54]. Moon et al. presented studies using relatively big datasets, i.e., 1650 sentences in 2020 [9] and then 4659 sentences in 2021 [8]. However, due to the scale-limited dataset, the semantic web and knowledge in some other forms are not versatile and comprehensive.

Though it can be concluded that the semantic web does provide assistance in the construction industry, especially when it is combined with the BIM environment. However, few studies are carried out specifically for the MEP engineering. On one hand, there are no relevant labeled datasets in the MEP domain for the deep learning IE implementation, thus current NER/RE methods for common domains based on deep learning cannot be adopted directly to the MEP domain. On the other hand, feature engineering, as well as efficient ways to carry out IE tasks or establish a proper dataset are not fully considered in most rule-based IE in the construction industry.

According to the above-mentioned problems, this paper proposes a novel rule-based approach to extract MEP information and establish an MEP semantic web. The approach firstly collects MEP corpora from the internet using a “snowball” strategy to establish a large-scale domain dataset. Then, NER based on suffix rules and RE based on dependency path rules are presented to improve the efficiency and accuracy of the IE tasks. Finally, the approach is adopted to build an MEP semantic web that consists of 15978 entities and 65110 relationship triples, proving its feasibility.

3 Methodology

The workflow of the proposed method is shown in Figure 1. To build an MEP semantic web, the MEP corpora should first be collected and preprocessed. Then, the proposed rule-based IE algorithms are applied to the corpora to match corresponding MEP entities and relationship triples. NER/RE outputs are further organized to generate the final MEP semantic web.

Basically, this study proposes an approach to extract MEP entities by suffix pattern rules from sentence segmentation outputs, and the relationship triples extraction is implemented by matching

dependency path rules. Moreover, to extract relationship triples as well as possible, and to discover some out-of-pattern triples, 2 ideas called “meta linking” and “path filtering” are proposed, respectively. Here the out-of-pattern triples refer to those meaningful and useful triples that cannot be matched by predefined pattern rules.

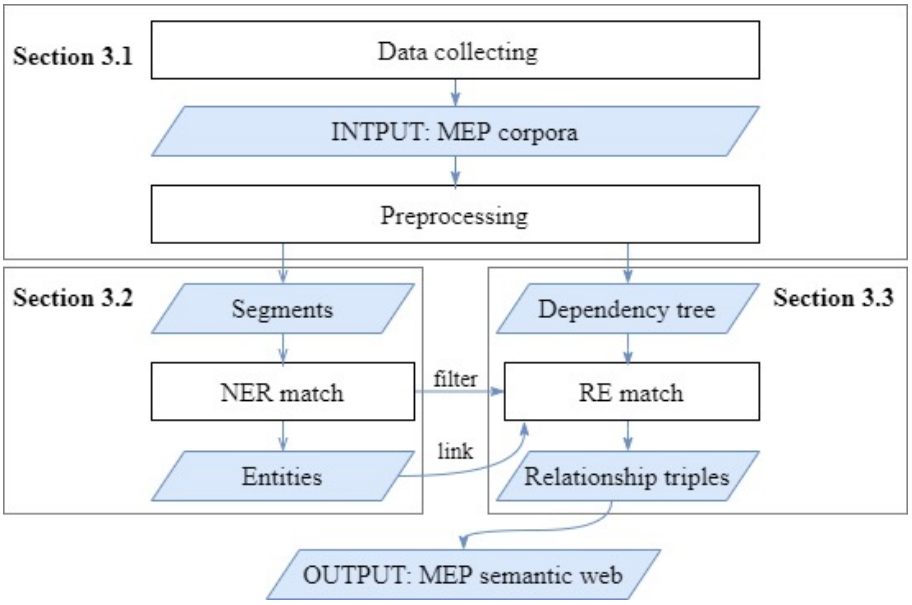

Figure 1. Overview of the proposed IE approach to establish an MEP semantic web

3.1 Corpora collecting and preprocessing

The first step of the approach is to collect MEP plain texts from the internet and generate the original MEP domain corpora. As mentioned before, the scale of the dataset is essential to build a satisfying semantic web. Therefore, to get as much data as possible, instead of sticking to specific data sources, the study proposed a “snowball” strategy to automate the finding process of the data source from the internet. In this process, the target data sources are not limited to design specifications but include technical manuals, engineering reports, research literature, etc. as well.

As shown in Figure 2, some essential websites are provided as seed data sources to extract original corpora in the form of seed keywords. The keywords extraction adopts the NER algorithms which will be discussed in section 3.2. Then, these keywords are considered as indexes for data retrieval and provided to portal sites’ search engine. The engines search the linkages within the websites and redirect to corresponding web page resources according to the indexes and linkages. Newly extended websites can be found and considered as new seed websites and thus the process can be repeated.. Normally, authoritative professional portal sites can be considered as the seed websites for the snowball expansion, like Wikipedia for the English language or Baidu for the Chinese language.

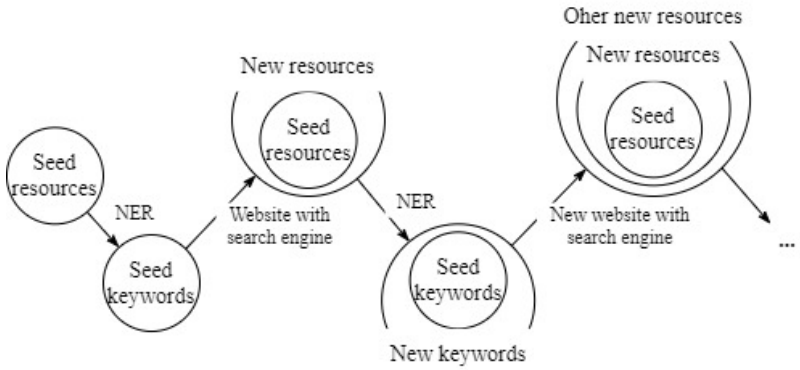

Figure 2. The snowball strategy for corpora collection

The corpora collecting process outputs a great number of texts that can be stored in database or data files. These plain texts are sent for preprocessing to support the subsequent procedures.

As shown in Figure 3, the preprocessing encompasses sentence split and some other basic NLP tasks including segmentation and dependency parsing. Some effective third-party NLP toolkits, such as *stanfordCoreNLP* [55] and *LTP* [56] have been developed to support this process. This study utilizes *LTP* as the upstream NLP toolkit to preprocess corpora and generate segmentations and dependency parsing results, which are the inputs of the following IE process.

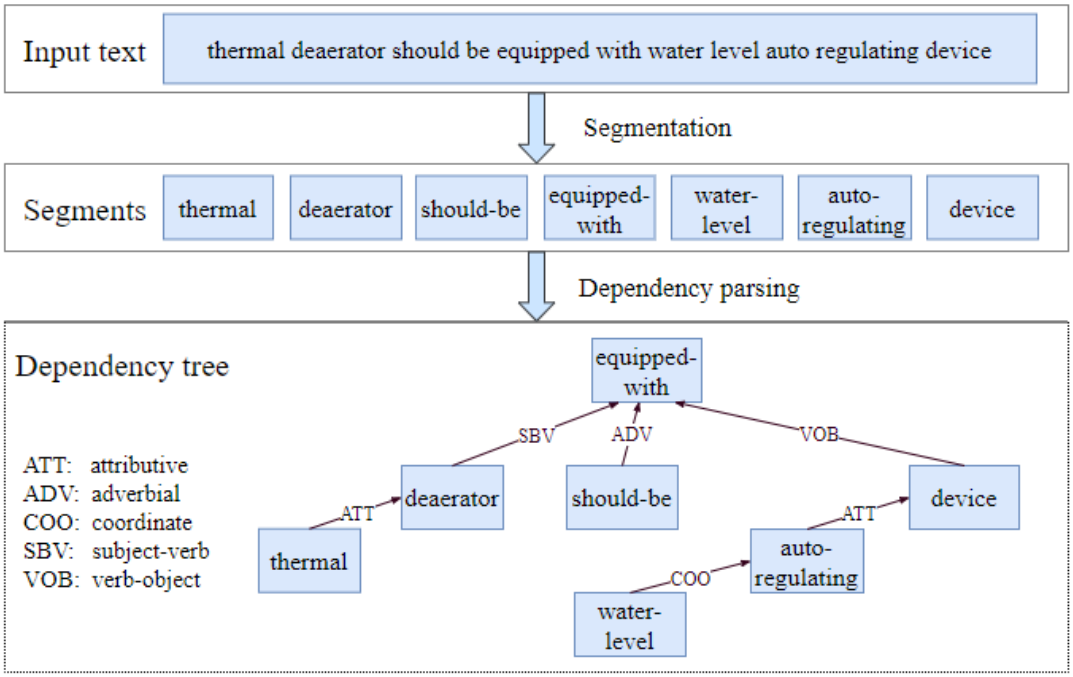

Figure 3. Segmentation/dependency parsing and corresponding outputs of the preprocessing

Segmentation in NLP refers to separating a sentence into a list of sub-strings. Each item of the list represents a relatively independent entity unit or syntactic unit, with lemmatization. Generally, sentences in English are separated with the blank space and thus easy to be separated. But still, lemmatization and words package are needed to get the meta-information of a sentence. However, there's no separator in a single sentence in Chinese or Japanese languages and so on. Therefore, the

segmentation process and toolkit are more important to these languages compared to English. Base on segmentation, dependency parsing can be further carried out to generate a tree structure called "dependency tree", where each node has a certain dependency relationship linking to its parent node. Figure 3 also gives an example about the process of segmentation and dependency parsing of an MEP knowledge recorded in a sentence in the form of natural language. The proposed method uses hyphens to connect related words as segments in segmentation, and then reconnect these segments into a dependency tree.

3.2 NER based on suffix rules

The workflow of the proposed NER solution is shown in Figure 4. First, the corpora are separated into a list of segments. During this sub-process, the times that a segment appears in a sentence or a paragraph should also be recorded. Then build a char suffix tree from this segments list and calculate the boundary entropy (BE) for every suffix within the tree. If the BE of a suffix is bigger than a given threshold, the suffix is considered as a candidate for succeeding procedures and added to the candidates list. These candidate suffixes are manually identified and classified to obtain the final suffix matching rules.

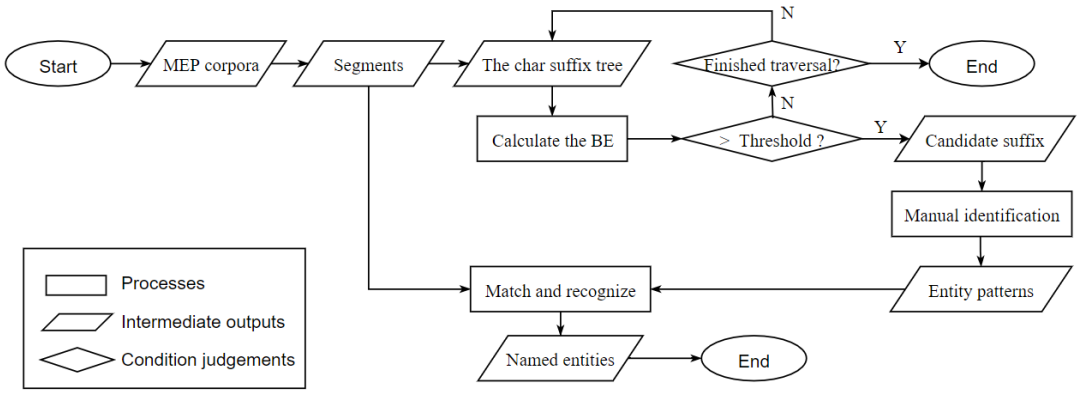

Figure 4. The workflow of the proposed NER solution

Algorithm 1 Matching Segments to the NER Patterns

```
for segment in segments:
    for pattern in patterns:
        if match(segment, pattern):
            entity_list[pattern].add(segment)
            break
```

Detailedly, entity phrases of the same category usually adopt different prefixes to represent specific identity, but the suffixes are mostly the same. According to this pattern and to match as many entities as possible with rule patterns, a matching method based on suffix patterns is proposed by analyzing the lexicon features of MEP entities. Notably, the suffix patterns can be generated from not only the last word of phrases, but also the sub-strings of the last word. Figure 5 shows a typical example of this "suffix of suffix" pattern, which can match both "motor" and "generator" by only

one suffix ".tor". Base on this idea, regular expressions can be designed for each entity class to distinguish whether the given segment belongs to a certain entity class or not.

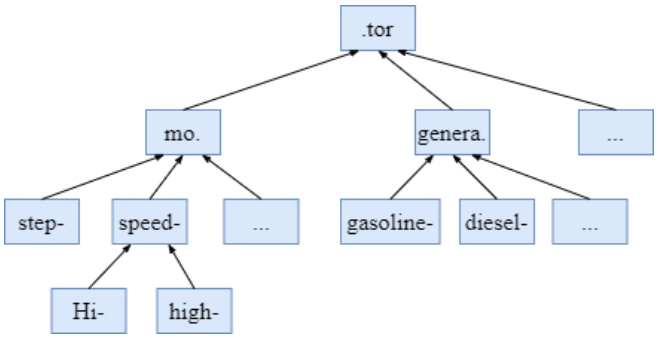

Figure 5. Same suffix with different prefixes, in which "." means a substring of a word.
A path from leaf node to root node generates a named entity, such as the "step-motor"

After segmentation, the matching method builds a character level suffix tree from all segments. Figure 6 is an example of a suffix tree generated from a list of four segments {monitor, barrier, system, mechanism}, where "mechanism" appears 1 time, "barrier" 2 times, "monitor" 3 times and "system" 5 times. Suppose freq(char) refers to the frequency of the character, then, freq("t"), freq("o") and freq("r") from "monitor" are both 3 as numbered in the figure. Similarly, freq("e") and freq("r") from "barrier" are 2. Thus, freq("tor") = 3 and freq("er") = 2, and the freq("r") from "er" and "tor" is 5. "$" refers to the end of the strings. In this segments list, "tor", "er", "tem" and "sm" can be 4 candidate suffixes. To evaluate which candidate is more appropriate and universal, the BE of each suffix can be calculated according to Eq (1) and Eq (2).

$$BE(suffix) = -\sum_{\substack{c\in \\ suffix\_pre\_char\_set}} P(c|suffix) \cdot log\, P(c|suffix) \qquad \text{Eq. (1)}$$

$$P(c|suffix) = \frac{P(\text{`}c\text{`}+\text{`}suffix\text{`})}{P(suffix)} \qquad \text{Eq. (2)}$$

where BE(suffix) is the boundary entropy of the suffix; P(suffix) is the probability of the appearance of the suffix, or the appearance number of the suffix in the total number of all suffixes. `c`+`suffix` means the new suffix generated by combining the letter 'c' and the suffix; suffix_pre_char_set refers the 'c' collection that existed in all `c`+`suffix` in the suffix tree.

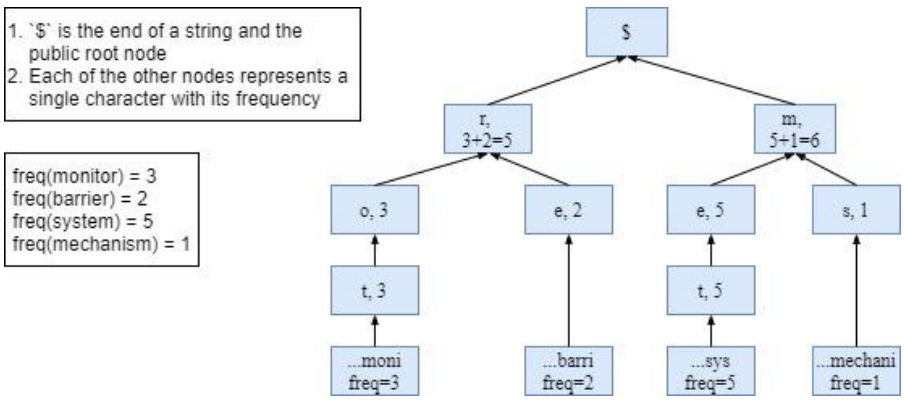

Figure 6. Example of character suffix tree

Higher BE means higher chaos between a suffix and its previous character set, indicating that the suffix is of higher generality. Hence, an appropriate BE threshold is able to find out general suffixes. However, it is an unsupervised process and there are no quantitative parameters to evaluate the threshold while the BE threshold is set to 8e-3 according to our tuning experiment. After this filtering process, a manual check is also suggested to ensure the rationality of these suffixes.

Finally, considering the characteristics of MEP engineering, the suffixes are manually classified into 7 categories, i.e., equipment, system, position, component/part, property, style, action/limitation. More details about these entity classes and suffix pattern examples are shown in Table 3. For example, the distributing-system is a named entity that recognized by the confirmed suffix "^.*tem$".

Table 3. detail and example about entity class and pattern

Class	Description	Entity example	Regex Pattern example
equipment	Ontology or instance about MEP equipment	High-speed-motor	^.*tor$
system	Common MEP system	Distributing-system	^.*tem$
position	Where the equipment/system is	Switching-room	^.*room$
Component/Part	Component or part/area of equipment	Valve-outlet	^.*outlet$
Property	Properties of an equipment or a system	Rated-voltage	^.*age$
Style	What instance it is or how it works	Cooled-option	^.*tion$
Action/Limitation	Operating action, or limitation of an instance of equipment or a system	Equipotential-bond	^.*bond$

Note: "^" and "$" refer to the start and the end of a string, "." Refers to any character or characters. "*" means that any times of the previous character can be matched, thus ".*" refers to any sub-string.

In general, the entity classes can describe most of the important information in MEP corpora, including the relevant characteristics, attributes, working conditions and operation methods of equipment. Meanwhile, this definition framework also reflects some relationship information that can be utilized in RE rules design that will be discussed later. Figure 7 shows some relationships related to the class "equipment". For example, if two named entities are extracted in a sentence and these two entities are classified into "equipment" and "place", a relationship between these two entities can be extracted as "where", referring that the "equipment" is located in the "place". The appendix enumerates all the 28 relationships defined by linking entity class pairs.

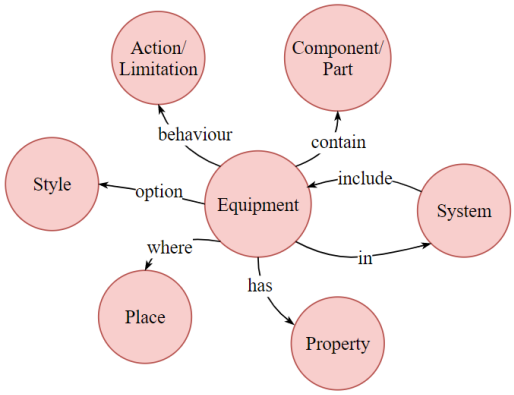

Figure 7. Some relationships defined by entity classes

Besides the accuracy and recall ratio of the recognized named entities, the length of entity phrases also has a significant influence on downstream tasks such as RE. For example, "*gasoline-generator*" is a legal and specific named entity, while "*gasoline*" and "*generator*" can also be two independent entities in some contexts. Apparently, it will be more difficult to generate relationships relevant to "*gasoline-generator*" because its appearances in sentences are much fewer compared to the separated two entities. Therefore, a "meta linking" idea is adopted in this study that the named entities are recognized as short as possible. Then, the longer-named entity in RE phase can be generated by linking several short entities together, like linking the "*gasoline*" and "*generator*" to build "*gasoline-generator*" in RE step.

3.3 RE based on dependency path rules

The workflow of the proposed RE solution is shown in Figure 8. First, the dependency types defined in LTP are translated into string patterns. Then establish the dependency tree by analyzing the dependency relationships within every sentence, followed by generating the dependency paths for every node pair in the dependency tree and translating the paths into strings as well. Next, regular matchings are carried out for the paths based on grammatical patterns. For those successfully matched paths, calculate the lengths of the paths and the proportion of entities. Only the paths that meet the threshold requirements are confirmed to be relationships between two entities, which are defined by connecting the start point and end point of the path. If the two entities have their own entity types, the relationship is defined according to their types, otherwise, defined by grammatical patterns.

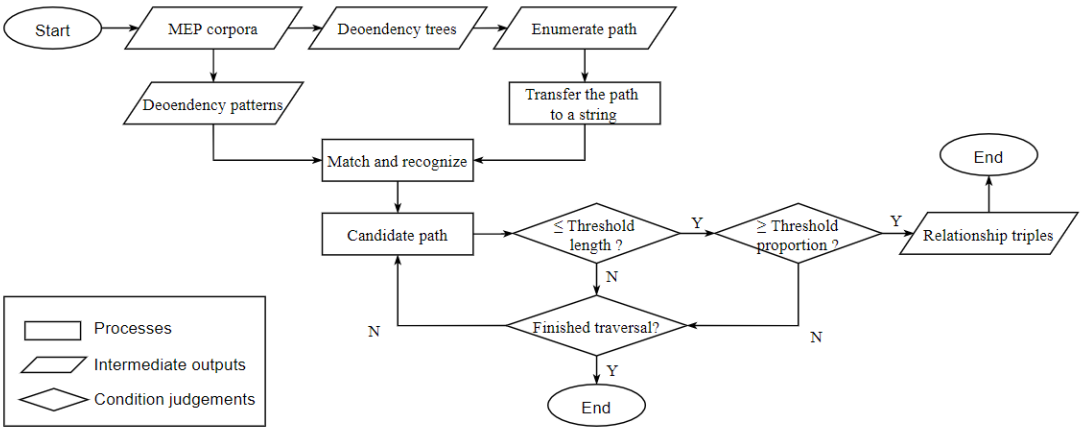

Figure 8. The workflow of the proposed RE solution

Algorithm 2 Matching Nodes in the Dependency Tree to the RE Patterns

```
for tree in denpendency_trees:
    for node1 in tree:
        for node 2 in tree and node2 != node 1:
            path = tree.get_path(node1, node2)
            pstr = path.to_string()
            for pattern in patterns:
                if match(pstr, pattern) and len(path) < max_len and entity_prop(path) > threshold:
                    triple_list[pattern].add((node1, node2))
                    break
```

Within this workflow, the RE rules are proposed to improve the accuracy of information extraction, based on the dependency tree. The dependency tree can be considered as an undirect graph, thus the paths between any two nodes in a node pair in the graph are unique. This path is named as dependency path in this study because it reflects syntactic information in the dependency tree. For a certain dependency path, the relationships on each edge are extracted and combined into a string representation. Take Figure 9 as an example, the path from node {*thermal*} to {*device*} emphasized by bold line can be represented as "ATT_SBV_VOB".

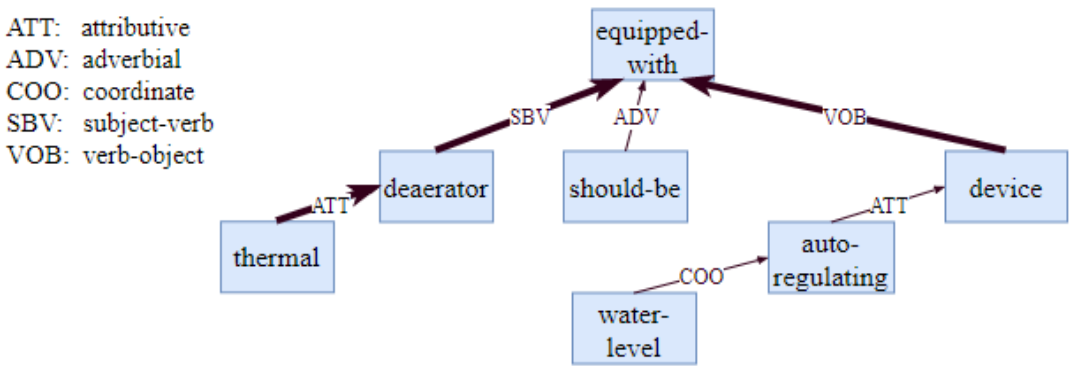

Figure 9. A dependency path example is emphasized by the bold line

The basic idea of RE matching is to retrieve the candidate entity pairs by dependency path matching, and then determine whether they have a relationship and what the relationship is. The proposed method enumerates nodes in the dependency tree in pairs, and then distinguishes whether their dependency path matches a pre-defined pattern. Because the dependencies are represented by strings, regular expressions are effective to describe and match dependency paths in the dependency tree to retrieve the candidate relationship chains. For example, the path shown in Figure 9 doesn't match any pre-defined patterns while the path from {*thermal*} to {*deaerator*} matches the pre-defined relationship {*thermal, attributive (style), deaerator*} as "^(ATT_)+$" listed in Table 4.

Table 4. Some patterns and relationships defined by dependency pattern

Pattern	Try to match	Example
^(SBV_)(\W*_)*(VOB)$	Subject-verb-object	"deaerator" →"equipped-with"(NO) →"device" Redefine as "subject-predicate"
^(ATT_)+$	A chain of attributive decoration	(*) →"thermal" (NO) →"deaerator" Redefine as "prefix attributive"
^(COO_)(\W*_)*(COO)$	Several things linked by "and", "or", etc.	"vessel" →(*) →"deaerator" Redefine as "cooperation"

Note1: "NO" means that it is not an MEP entity; "(*)" means an arbitrary number (including zero) of segments

Note2: Symbols in regular patterns represent the same meaning as Table 3.

Here are some more details about RE match strategies in this paper.

(1) Filtering strategy. Because the regular expressions only capture the syntactic information without semantic information about MEP, thus all candidate pairs that match certain patterns need to be filtered to ensure the data quality, by the following 2 concerns. 1) Overlong paths are dropped, because if the dependency path is too long, that is, the start node and end node are too far away in the dependency tree, there is a high possibility that the two nodes are not related. 2) Paths containing few MEP entities are filtered because too many non-MEP words will bring noises to the outputs. The NER match rules can be used for this filter idea, and it is named as "path filtering" in this study. According to our tuning experiment, the threshold configuration is set as follows:

a) Overlong path: contains more than 5 nodes, or

$$Len(path) = Freq(node \mid node \in path) > 5 \quad \text{Eq. (3)}$$

b) Few MEP words: the proportion of MEP nodes is less than 1/2, or

$$Prop = \frac{Freq(node_{MEP} \mid node_{MEP} \in path)}{Len(path)} < 1/2 \quad \text{Eq. (4)}$$

(2) RE matches based on class definition. Most of the relationship types are generated by connecting entity classes defined in Table 3. In most cases, it is sufficient to indicate the relationship by connecting entity classes. There are totally 28 relationships defined by entity classes connection, and some typical relationships are shown in Table 5.

(3) RE matches based on class definition dependency patterns. Because a small number of non-MEP words are allowed to be involved in RE outputs, it is possible that some relationships cannot be defined by linking two entity classes. In this case, the pre-defined dependency patterns as shown in Table 4 and Table 6 can be assigned as the relationship of these two entities.

Table 5. Some basic relationship classes defined by entity class connection

Relationship Prompt	Explanation	Example
Equipment-Place	Where the equipment is	*"generator"→"power station"*
System-Equipment	Contain	*"transformer system"→ "transformer"*
Property-Style	Specific description	*"DC"→"voltage"*
Equipment-Property	Equipment has a certain property	*"motor"→"voltage"*
Property-Action/Limitation	Limitation of physical quality	*"voltage"→"no greater than"*

Table 6. Dual dependency patterns defined by LTP

^(SBV_)(\W*_)$	^(\W*_)(VOB_)$	^(ADV_)(\W*_)$	^(FOB_)(\W*_)$
^(CMP_)(\W*_)$	^(COO_)(\W*_)$	^(ATT_)(\W*_)$	^(POB_)(\W*_)$
^(LAD_)(\W*_)$	^(RAD_)(\W*_)$	^(DBL_)(\W*_)$	^(IOB_)(\W*_)$

Note1: The explanations of the short forms are listed as follows. SBV: subject-verb; VOB: verb-object; ADV: adverbial; FOB: fronting-object; CMP: complement; COO: coordinate; ATT: attribute; POB: preposition-object; LAD: left adjunct; RAD: right adjunct; DBL: double; IOB: indirect-object.

Note2: Symbols in regular patterns represent the same meaning as in Table 3.

Points (2) and (3) also give us a good side-effect, which allows the rule-based method to discover relationship information out of the predefined rules, though these dependency pattern relationships may contain noise because they allow the existence of non-MEP words.

4 Case Study

In general, an MEP semantic web that contains 15978 entities and 65110 relationship triples has been established from MEP corpora in this case study, with a precision of about 81% to entities and 75% to relationship triples, respectively. Details on the methodology implementation and application are described as follows, including data collecting, rule design, algorithm runtime analysis and output analysis. The last two analyses illustrate some interesting information about MEP corpora and the semantic web, which will be discussed later.

(1) **Data collecting and preprocessing**. A total of 65MB (over 270,000 sentences) of Chinese corpora were collected from the internet, including design specifications (19MB), academic papers (7MB), web encyclopedias (11MB) and discussions in forums (28MB). Crawler programs were developed with the environment of python3 and dependency of selenium-python (an open-source package to manipulate the browser at code level). In detail, 1) 200 academic literatures were downloaded manually. MEP design specifications were collected from *www.jianbiaoku.com* by crawler program. Then the seed corpora were built based on these two data sources. 2) Preprocessing was applied including analyzing the corpora lexicon and segmentation features, and extracting 4218 keywords (Table 7) such as energy-saving, rated-voltage, air-conditioning-system, etc. as the original entity keywords. 3) *baike.baidu.com* and *zhidao.baidu.com* were selected as target portal sites and then corresponding crawler programs were developed according to the contents of these websites. Then the MEP corpora from the web encyclopedias and forums were collected to enrich the final corpora.

(2) **Rule design**. After the data collecting was accomplished, a total number of 91 NER rules,

in which some of the examples are listed in Table 3, were designed according to corpora lexicon and segmentation features. On the other hand, 15 patterns in total were defined for RE rules, including 12 dual dependency patterns defined by *LTP* in Table 6 and 3 multiple dependency patterns listed in Table 4, i.e., subject-predicate-object, prefix attributive decoration, cooperation. Table 7 shows the exact numbers of these patterns.

Table 7. Detail about rule match

	Entity							Triple
Class	equipment	system	place	component/part	property	style	Action/limit	overall
# original keywords	2051	446	126	171	989	184	251	4218
# pattern	23	6	9	11	17	5	20	15
# output	4456	973	879	1409	3343	3216	1702	65110
Precision	82%	84%	72%	74%	84%	92%	72%	75%

Note 1: "#" means number. For each column of entity, 100 samples are randomly selected and the proportion of correct samples is then calculated as precision. For triple, 200 samples are randomly selected to calculate the precision in the same way.

Note 2: Precision of triple contains both the error of NER and RE since the RE depends on NER.

Note 3: The triple's precision is evaluated without considering the difference between classes.

(3) **Algorithm running**. It took about 1 hour to accomplish the IE process on the 65MB corpora in this study. As shown in Figure 10, it can be concluded that the extraction speed of the proposed rule-based algorithm is stable without significant attenuation, indicating that a stable proportion of new entities/triples can be extracted when new text data is inputted. Therefore, it is reasonable to expect that much more entities/triples can be extracted if more corpora are collected. Another conclusion from Figure 10 is that the connection of entities is relatively sparse in MEP domain. A dense graph requires the number of edges to be the square times of the number of nodes while Figure 10 shows that the number of edges is only a linear constant time to the number of nodes. Hence, it is also reasonable to speculate that almost all MEP sentences are carrying the similar amount of information.

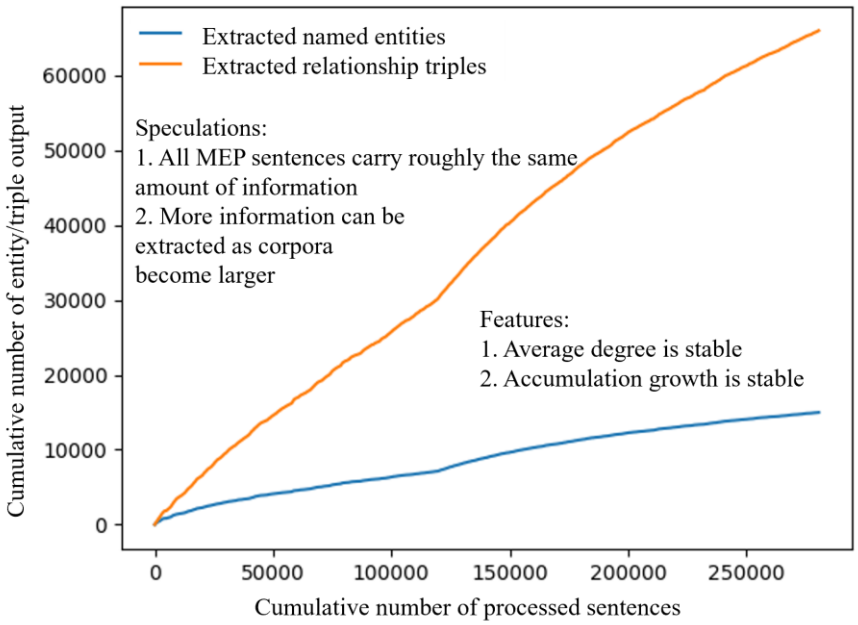

Figure 10. Entity/triple accumulation as algorithm processing

(4) **Algorithm outputs**. Table 7 also shows the statistics of the output that totally 15978 entities and 65110 relationship triples had been extracted. The scale of the NER results is almost the same as ACE-2004 NER datasets mentioned in Table 2, and the scale of the RE results is much greater than ACE-2004 and TACRED RE datasets, since a relationship triple corresponds to at least one sentence. The number of the labeled sentences will increase rapidly if the distant supervision algorithm was applied. On the other hand, the proposed rule match algorithm also reaches the average precision, which is about 81% for both NER and RE, as mentioned in section 2.3. Besides, because the proposed algorithm deals with both NER and RE tasks at the same time, the precision is better than the NER-RE joint solutions based on deep learning in Table 1. Moreover, Table 8 shows the connectivity statistics of the MEP semantic web generated by the proposed rule match algorithm. It can be concluded that the semantic web has strong connectivity since it contains a large subgraph with a size of 14342 entity nodes, which accounts for nearly 90% of all entities. Figure 10, together with Figure 11, illustrate that the whole semantic web is highly centralized, indicating that the MEP domain has high coupling characteristic.

Table 8. Connectivity statistics of the established semantic web

Subgraph size	Max degree	Node with max degree
14342	1572	system
16	13	terminal
6	4	check
4	3	plug
Another 23 graph with size = 3	--	--
Another 267 graph with size = 2	--	--

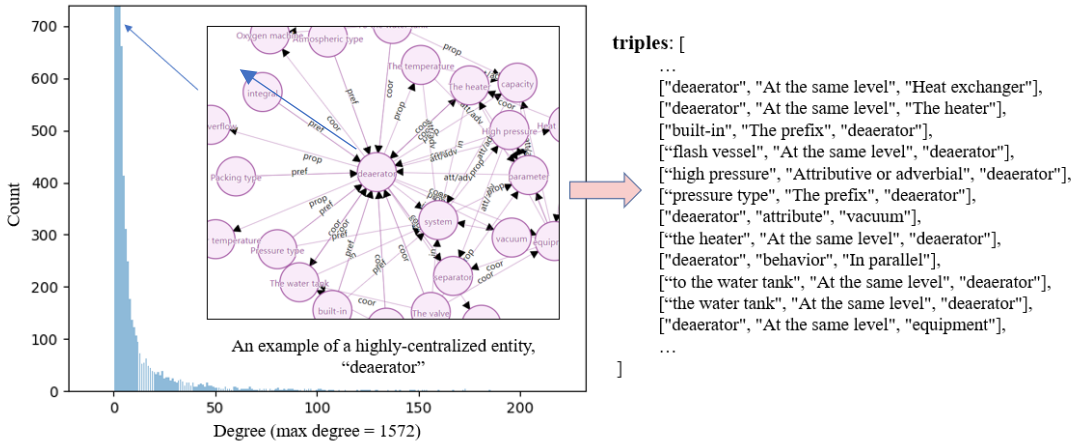

Figure 11. Degree distribution of the established semantic web and a highly-concentrated named entity example

5 Discussions

5.1 In comparison with IE based on deep learning models

As mentioned previously, current deep learning models for common domain IE are not satisfying in the aspect of performance in the minority language and specific professional domain. To prove this declaration, labeled datasets were constructed for the training and evaluation of some classical NER/RE models, by the proposed rule match algorithm. 70% of sentences were utilized with their corresponding entities/triples outputs to construct the labeled samples for NER/RE models training, tuning and testing. Then, these NER/RE models were applied to predict new entities/triples on the rest of 30% corpora (just ignored the rule-based output), with manual precision evaluation, as the practical prediction.

In detail, the following deep learning models were trained for this comparison experiment: 1) Non-class LSTM [25] + BERT classification for NER. 2) Residual CNN [35] for RE. Table 9 shows some statistics of the deep learning models and the proposed algorithm, and 2 conclusions can be obtained.

(1) Single deep learning model of NER/RE cannot reach the expected performance without other measures. Supervised learning is under the assumption that all samples in the dataset are correct. However, according to the manual evaluation, these two deep learning models cannot achieve enough precisions, even though these precisions are close to their performances in other NER/RE tasks of general domains as shown in Table 1.

(2) Deep learning models performed even worse in practical prediction than in test datasets. The reason may be that the datasets used for model training and testing are not covering all the patterns, and with unavoidable errors while at the same time error propagation has a significant impact on the performance of the model. Take NER as an example, the precision of the model is only 51%, much worse than the proposed rule match solution, whose precision is 81%.

Table 9. Performance overview

Task	Method	Machine test best case		Machine test worst case		Manual
		Class	Precision	Class	Precision	Macro Precision*
NER	LSTM +BERT	equipment	95.77%	Component/ Part	13.89%	51%
	Rule match	--	--	--	--	81% (Improvement=37%)
RE	CNN	Equip-prop	68.74%	Style-place	12.82%	38%
	Rule match	--	--	--	--	75% (Improvement=49%)

Note: Macro precision doesn't care about which class the sample belongs to.

Improvement = 1 - Precision_{DL} / Precision_{RULE}

Precision = Count (correct samples) / Count (all samples)

The reasons why the deep learning models in practical simulation could not reach the expected results, or in other words, could not fit the datasets well, are summarized as follows.

(1) Mislabeling problem. Some instances may be out of the testing patterns. This problem causes the assumption of independence and identical distribution to be invalid. However, deep learning models capture and recognize these patterns in high-dimensional hidden vector space which cannot be understood easily.

(2) Long-tail problem. Some classes have few instances, thus making the deep learning models prefer classes with a large number of instances to the others. The impact of this problem can be alleviated by over-sampling or under-sampling strategies.

(3) Transferring problem. The BERT embedded in NER and RE models was pre-trained by common domain corpora, which may not capture the features of MEP field texts effectively. This problem is expected to be addressed by applying fine-tune on BERT (or other pre-trained models).

(4) Limitation of deep-learning-based NLP model itself. Natural language is more complex and difficult to process compared with structured data, such as data in computer vision. But the BERT and LTP utilized in this study are far from perfect and still make mistakes. These issues may also affect downstream tasks of NER and RE.

5.2 Future directions

In general, this paper proposes a rule-based approach to extract information from the professional MEP domain, with both high execution efficiency and high IE accuracy. The performance of the proposed approach is also satisfying, especially when compared with the IE methods based on deep learning. However, the rule-based IE approach still suffers a limit and has the potential to be improved, including the following points.

(1) **Scalability**. The biggest problem and technique difficulties that all rule-based methods face is scalability, since these methods can hardly find out useful information when the input is out of the predefined patterns. Some scalable strategies adopted in this study, such as fuzzy entity match by suffix in NER, and the dependency path filter in RE are proved to be feasible to partially solve this problem. But this limitation still hinders the discovery of brand new NER/RE patterns. New ideas to improve scalability are needed in the aspect of both strategy and algorithm.

(2) **Scale**. According to the conclusion obtained in the case study, lots of MEP entities/triples

can be extracted from corpora with steady growth, and it is reasonable to expect that much more entities/triples can be extracted if the corpora continue to scale up. Hence, a more effective corpora collecting strategy should be proposed. Besides, another possible way to scale up the semantic web is to extract more entities/triples by more patterns from the existed corpora. These patterns can be either predefined or discovered by algorithms.

(3) **Class breakdown**. MEP domain encompasses many sub-domains, which can be further divided into numerous specific classes. For example, "generator" and "motor" can be considered as 2 more specific classes in the electricity domain. Predefining more entity classes should be helpful to find accurate knowledge. Some statistics machine learning methods like clustering are also expected to be feasible to deal with this problem.

(4) **Dealing with ambiguity and co-referential**. Ambiguity means that an entity string may represent different semantic meanings in various contexts, while co-referential means that some different entity strings may refer to the same thing. Current NER/RE match strategies cannot solve these two problems, which however significantly affect the practical utilization of the semantic web. Referring to the concept of the buildingSMART Data Dictionary (bSDD), a library of object concepts and their properties that embedded in the solution may be a future direction to find the right classifications, properties and values within the text contents.

(5) **Languages**. No universal solution to deal with different languages is one common challenge for both IE and NLP. In this study, the collected corpora are mostly in Chinese while at the same time, English corpora have also been collected marginally to test the whole approach. Fortunately, in the proposed approach, language itself is not a big problem because several mature toolkits have been developed to carry out the general NLP tasks. The *stanfordCoreNLP* was adopted for English corpora and *LTP* for Chinese corpora. They both worked well to support the succeeding process and proved that the framework of the proposed approach is feasible regardless of language. However, it still takes efforts to deal with suffixes and specific patterns in different languages manually. This problem should be expected to be solved when language translation by artificial intelligence algorithms becomes more accurate.

6 Conclusion

The MEP domain has accumulated a lot of knowledge and experiences represented in natural language texts. However, these text-based materials are difficult to be processed and utilized with traditional methods for a long time. Recently, the semantic web and KG have been considered as promising technology to utilize information in text forms, once they had been constructed by IE. However, there is no common but effective way to establish a semantic web for professional domains, especially for a certain domain (e.g., the MEP domain) with complex domain knowledge and little labeled data.

With the segmentation and dependency parsing output by third-party toolkits, this paper proposed a novel approach to extract useful information and build a semantic web for the MEP domain with hybrid strategies. (1) A "snowball" strategy to collect large-scale MEP corpora. (2) For NER, a suffix match strategy to distinguish whether the certain segment is an MEP entity or not, with classification information. (2) For RE, a dependency path match strategy to distinguish whether the certain segment/entity pair has a syntactic relationship or not, and to clarify what the relationship

is. (3) The filter/accept strategy for dependency path in RE is also proposed to discover extra useful entities and relationships which are out of the predefined patterns. According to the proposed rule-based approach, an original MEP semantic web, with about 15000 entities and 65000 relationships has been established and the extraction precision of the entities and relationships are 81% and 75%, respectively. Furthermore, a comparison experiment between classical deep learning models and the proposed rule-based approach was carried out as well, and it proved that the performance of the proposed approach maintained about 37% and 49% improvement of the precision, compared with deep learning NER and RE, respectively. In general, the case study and the comparison experiment proved the necessity and feasibility of the proposed rule-based approach.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 51778336), and the Tsinghua University – Glodon Joint Research Center for Building Information Modeling. The LTP utilized in this study is supported by the Social computing and information retrieval research center of Harbin Institute of Technology (HIT-SCIR) and the stanfordCoreNLP is supported by the Stanford NLP Group.

References

[1] J Cowie, W Lehnert. "Information Extraction." Communications of the ACM 39.1(1996), pp. 80-91, https://doi.org/10.1145/234173.234209.

[2] Shadbolt N, W Hall, and T. Berners-Lee. "The Semantic Web Revisited." IEEE Intelligent Systems 21.3(2006), pp. 96-1011, https://doi.org/10.1109/MIS.2006.62.

[3] Pujara J, Miao H, Getoor L, et al. "Knowledge Graph Identification." Springer-Verlag New York, Inc. (2013), pp. 542-5571, https://doi.org/10.1007/978-3-642-41335-3_34.

[4] Chen W, Chen K, Cheng J, et al. "BIM-based framework for automatic scheduling of facility maintenance work orders." Automation in Construction 91 (2018), pp. 15-301, https://doi.org/10.1016/j.autcon.2018.03.007.

[5] Zhang J, El-Gohary N M. "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." Journal of Computing in Civil Engineering 30.2(2016), p. 1410130644410001, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346.

[6] Zhang J, El-Gohary N M. "Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking." Automation in Construction 73 (2016), pp.45-571, https://doi.org/10.1016/j.autcon.2016.08.027.

[7] Amor R, Dimyadi J. "The Promise of Automated Compliance Checking." Developments in the Built Environment 5 (2020), p. 1000391, https://doi.org/10.1016/j.dibe.2020.100039.

[8] Moon S, Lee G, Chi S, et al. "Automated Construction Specification Review with Named Entity Recognition Using Natural Language Processing." Journal of Construction Engineering and Management 147.1(2021), p. 04020147 1, https://doi.org/10.1061/(ASCE)CO.1943-

7862.0001953.

[9] Moon S, Chung S, Chi S. "Bridge Damage Recognition from Inspection Reports Using NER Based on Recurrent Neural Network with Active Learning." Journal of Performance of Constructed Facilities 34.6(2020), p. 04020119, https://doi.org/10.1061/(ASCE)CF.1943-5509.0001530.

[10] Woo J H, Clayton M J, Johnson R E, et al. "Dynamic Knowledge Map: reusing experts' tacit knowledge in the AEC industry." Automation in Construction 13.2(2004), pp. 203-207, https://doi.org/10.1016/j.autcon.2003.09.003.

[11] Williams T P, J F Betak. "Identifying Themes in Railroad Equipment Accidents Using Text Mining and Text Visualization." International Conference on Transportation & Development(2016), pp. 531-537, https://doi.org/10.1061/9780784479926.049.

[12] Xue J, Shen G Q, Li Y, et al. "Dynamic Analysis on Public Concerns in Hong Kong-Zhuhai-Macao Bridge: A Topic Modeling Approach." Construction Research Congress(2020), pp. 160-170, https://doi.org/10.1061/9780784482889.018.

[13] Wang Y, Taylor J E. "DUET: Data-Driven Approach Based on Latent Dirichlet Allocation Topic Modeling." Journal of Computing in Civil Engineering 33.3(2019), p. 04019023.1-04019023.8, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000819.

[14] Mostafa Ali, Yasser Mohamed. "A method for clustering unlabeled BIM objects using entropy and TF-IDF with RDF encoding." Advanced Engineering Informatics 33 (2017), pp. 154-163, https://doi.org/10.1016/j.aei.2017.06.005.

[15] Paulheim, Heiko. "Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods." Semantic Web 8.3(2017), pp. 489-508, https://doi.org/10.3233/SW-160218.

[16] Terkaj W, Šojić A. "Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology." Automation in Construction 57(2015), pp. 188-201, https://doi.org/10.1016/j.autcon.2015.04.010.

[17] Gómez-Romero J, Bobillo F, Ros M, et al. "A fuzzy extension of the semantic Building Information Model." Automation in Construction 57(2015), pp. 202-212, https://doi.org/10.1016/j.autcon.2015.04.007.

[18] Kim K, Kim H, Kim W, et al. "Integration of ifc objects and facility management work information using Semantic Web." Automation in Construction 87(2018), pp. 173-187, https://doi.org/10.1016/j.autcon.2017.12.019.

[19] Ding L Y, Zhong B T, Wu S, et al. "Construction risk knowledge management in BIM using ontology and semantic web technology - ScienceDirect." Safety Science 87(2016), pp. 202-213, https://doi.org/10.1016/j.ssci.2016.04.008.

[20] Werbrouck J, Pauwels P, Bonduel M, et al. "Scan-to-graph: Semantic enrichment of existing building geometry." Automation in Construction 119(2020), p. 103286, https://doi.org/10.1016/j.autcon.2020.103286.

[21] Patacas J, Dawood N, Greenwood D, et al. "Supporting building owners and facility managers in the validation and visualisation of asset information models (AIM) through open standards and open technologies." Electronic Journal of Information Technology in Construction 21(2016), pp. 434-455, https://www.itcon.org/paper/2016/27, (accessed on Oct. 31st, 2021).

[22] Kalasapudi V S, Turkan Y, Tang P. "Toward Automated Spatial Change Analysis of MEP Components Using 3D Point Clouds and As-Designed BIM Models." 2014 2nd International Conference on 3D Vision 2(2014), pp. 145-152,

https://doi.org/10.1016/10.1109/3DV.2014.105.

[23] Beach T H, Rezgui Y, Li H, et al. "A rule-based semantic approach for automated regulatory compliance in the construction sector." Expert Systems with Applications 42.12(2015), pp. 5219-5231, https://doi.org/10.1016/10.1016/j.eswa.2015.02.029.

[24] Zhang J, El-Gohary N M. "Semantic-Based Logic Representation and Reasoning for Automated Regulatory Compliance Checking." Journal of Computing in Civil Engineering 31.1(2017), p. 4016037.1, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000583.

[25] Lample G, Ballesteros M, Subramanian S, et al. "Neural Architectures for Named Entity Recognition." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2016), pp. 260-270, https://arxiv.org/abs/1603.01360v2, (accessed on Oct. 31st, 2021).

[26] Chiu J P C, Nichols E. "Named Entity Recognition with Bidirectional LSTM-CNNs." Computer Science (2015), https://doi.org/10.1162/tacl_a_00104.

[27] Zukov-Gregoric A, Bachrach Y, Minkovsky P, et al. "Neural Named Entity Recognition Using a Self-Attention Mechanism." 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI) IEEE (2017), pp. 652-656, https://doi.org/10.1109/ICTAI.2017.00104.

[28] Devlin J, Chang M W, Lee K, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." (2018), https://arxiv.org/abs/1810.04805, (accessed on Oct. 31st, 2021).

[29] Radford A, Narasimhan K, Salimans T, et al. "Improving language understanding by generative pre-training."(2018). https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf, (accessed on Oct. 31st, 2021).

[30] Ma R, Peng M, Zhang Q, et al. "Simplify the Usage of Lexicon in Chinese NER." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2019), pp. 5951-5960, https://doi.org/10.18653/v1/2020.acl-main.528.

[31] Zhou Y, Zheng X, Huang X. "Chinese Named Entity Recognition Augmented with Lexicon Memory." (2019), https://arxiv.org/abs/1912.08282, (accessed on Oct. 31st, 2021).

[32] Wu Z, Yu Z, Guo J, et al. "Fusion of Long Distance Dependency Features for Chinese Named Entity Recognition Based on Markov Logic Networks." 1st CCF Conference on Natural Language Processing and Chinese Computing (2012), pp. 132-142, https://doi.org/10.1007/978-3-642-34456-5_13.

[33] Wei Z, Su J, Wang Y, et al. "A Novel Cascade Binary Tagging Framework for Relational Triple Extraction." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020), pp. 1476-1488, https://arxiv.org/abs/1909.03227, (accessed on Oct 31st, 2021).

[34] Xu H, Wang W, Mao X, et al. "Scaling up Open Tagging from Tens to Thousands: Comprehension Empowered Attribute Value Extraction from Product Title." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019), pp. 5214-5223, https://doi.org/10.18653/v1/P19-1514.

[35] Zeng D, Liu K, Chen Y, et al. "Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks." Conference on Empirical Methods in Natural Language Processing (2015), pp. 1753-1762, https://doi.org/10.18653/v1/D15-1203.

[36] Lin Y, Shen S, Liu Z, et al. "Neural Relation Extraction with Selective Attention over Instances." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics 2016 (Volume 1: Long Papers) (2016), pp. 2124-2133, https://doi.org/10.18653/v1/P16-1200.

[37] Li Q, Ji H. "Incremental Joint Extraction of Entity Mentions and Relations." Meeting of the Association for Computational Linguistics 2014 (Volume 1: Long Papers) (2014), pp. 402-412, https://doi.org/10.3115/v1/p14-1038.

[38] Zheng S, Wang F, Bao H, et al. "Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2017), pp. 1227-1236, https://doi.org/10.18653/v1/P17-1113.

[39] Mintz M, Bills S, Snow R, et al. "Distant supervision for relation extraction without labeled data." ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (2009), pp. 1003-1011, https://doi.org/10.3115/1690219.1690287.

[40] Riedel S, Yao L, McCallum A. "Modeling relations and their mentions without labeled text." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg (2010), pp. 148-163, https://doi.org/10.1007/978-3-642-15939-8_10.

[41] Feng J, Huang M, Zhao L, et al. "Reinforcement learning for relation classification from noisy data." Proceedings of the AAAI Conference on Artificial Intelligence 32.1(2018), pp. 5779-5786, https://ojs.aaai.org/index.php/AAAI/article/view/12063, (accessed on Oct. 31st, 2021).

[42] Qin P, Xu W, Wang W Y. "Dsgan: Generative adversarial training for distant supervision relation extraction." 56th Annual Meeting of the Association-for-Computational-Linguistics (ACL), pp. 496-505, https://doi.org/10.18653/v1/P18-1046.

[43] Deng J, Cheng L, Wang Z. "Self-attention-based BiGRU and capsule network for named entity recognition." (2020), https://arxiv.org/abs/2002.00735, (accessed on Oct. 31st, 2021).

[44] Kim J H and P C Woodl. "A rule-based named entity recognition system for speech input." Sixth International Conference on Spoken Language Processing 1(2000), pp. 528-531, https://www.isca-speech.org/archive_v0/archive_papers/icslp_2000/i00_1528.pdf, (accessed on Oct. 31st, 2021).

[45] Collins M, Singer Y. "Unsupervised models for named entity classification." 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999), https://aclanthology.org/W99-0613.pdf, (accessed on Oct. 31st, 2021).

[46] Huffman S B. "Learning information extraction patterns from examples." International Joint Conference on Artificial Intelligence (1995), pp. 246-260, https://doi.org/10.1007/3-540-60925-3_51.

[47] Mooney R. "Relational learning of pattern-match rules for information extraction." Proceedings of 1998 Spring Symposium Series Applying Machine Learning to Discourse Processing (1998), pp. 6-11, https://www.webofscience.com/wos/alldb/full-record/INSPEC:6408594, (accessed on Oct. 31st, 2021).

[48] Shaalan K. "Rule-based approach in Arabic natural language processing." The International Journal on Information and Communication Technologies (IJICT) 3.3 (2010), pp. 11-19, http://www.ieee.ma/IJICT/IJICT-SI-Bouzoubaa-3.3/3%20-%20Khaledl.pdf, (accessed on Oct.

31st, 2021).

[49] Pomares Quimbaya A, Sierra Múnera A, González Rivera R A, et al. "Named Entity Recognition Over Electronic Health Records Through a Combined Dictionary-based Approach." Procedia Computer Science 100(2016), pp. 55-61, https://doi.org/10.1016/j.procs.2016.09.123.

[50] Jiang M, Shang J, Cassidy T, et al. "Metapad: Meta pattern discovery from massive text corpora" Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017), pp. 877-886, https://doi.org/10.1145/3097983.3098105.

[51] Hanisch D, Fundel K, Mevissen H T, et al. "ProMiner: rule-based protein and gene entity recognition." BMC Bioinformatics 6.1 (2005), pp. 1-9, https://doi.org/10.1186/1471-2105-6-S1-S14.

[52] Rahem, K.R. and N. Omar, "Rule-based named entity recognition for drug-related crime news documents." Journal of Theoretical and Applied Information Technology 77.2(2015), pp. 229-235, https://www.webofscience.com/wos/alldb/full-record/INSPEC:15842464, (accessed on Oct. 31st, 2021) .

[53] Zhang, R. and N. El-Gohary, "A Machine-Learning Approach for Semantically-Enriched Building-Code Sentence Generation for Automatic Semantic Analysis." Construction Research Congress (2020), pp. 1261-1270, https://doi.org/10.1061/9780784482865.133.

[54] Chi N W, Lin K Y, El-Gohary N, et al. "Gazetteers for Information Extraction Applications in Construction Safety Management." ASCE International Workshop on Computing in Civil Engineering (2017), pp. 401-408, https://doi.org/10.1061/9780784480847.050.

[55] StanfordCoreNLP. https://stanfordnlp.github.io/CoreNLP, (accessed on Oct. 31st, 2021).

[56] Che W, Li Z, Liu T. "LTP: A Chinese Language Technology Platform." The 23rd International Conference on Computational Linguistics (2010), pp. 13-16, https://aclanthology.org/C10-3004.pdf, (accessed on Oct. 31st, 2021).

Glossary

IE: information extraction
MEP: mechanical, electrical and plumbing
NER: named entity recognition
RE: relation extraction
EE: event extraction
KG: knowledge graph
IR: information retrieval
Q&A: question answering
HVAC: heating, ventilation and air-conditioning
FM: facility management
AEC/FM: architecture, engineering, construction, and facility management
BIM: building information modeling
IFC: industrial foundation class
OWL: ontology web language
RDF: resource description framework

Appendix

28 relationship types define by entity classes linking:

Index	Head entity class	Tail entity class	Relationship
1	equipment	equipment	Equipment
2	equipment	system	Equipment in a system
3	equipment	position	Equipment in a place
4	equipment	component/part	Component of an equipment
5	equipment	property	Property of an equipment
6	equipment	style	Style of an equipment
7	equipment	action/limit	Action or limitation of an equipment
8	system	system	Systems
9	system	position	System in a place
10	system	component/part	Component of a system
11	system	property	Property of a system
12	system	style	Style of a system
13	system	action/limit	Action or limitation of a system
14	position	position	Positions
15	position	component/part	A component in a place
16	position	property	Property of a position
17	position	style	Description of a position
18	position	action/limit	Actions in or limitation of a position
19	component/part	component/part	components
20	component/part	property	Property of a component
21	component/part	style	Style of a component
22	component/part	action/limit	Action or limitation of a component
23	property	property	Properties
24	property	style	Description of a property
25	property	action/limit	Property related to action or limitation
26	style	style	Styles
27	style	action/limit	Style of an action
28	action/limit	action/limit	Actions or limitations

15 relationship types defined by dependency pattern are shown in Table 4 and Table 6.