Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Syntactic and semantic information extraction from NPP procedures utilizing natural language processing integrated with rules

Yongsun Choi [a, *], Minh Duc Nguyen [b], Thomas N. Kerr Jr. [c]

[a] *Department of Industrial and Management Engineering, Inje University, 197 Inje-ro, Gimhae, Gyeongnam 50834, Republic of Korea*
[b] *Department of Information and Communication Systems, Inje University, 197 Inje-ro, Gimhae, Gyeongnam 50834, Republic of Korea*
[c] *Department of Operations Procedures, James A. FitzPatrick NPP, 268 Lake Rd, Oswego, NY 13126, USA*

ARTICLE INFO

Article history:
Received 19 June 2020
Received in revised form
5 August 2020
Accepted 9 August 2020
Available online 19 August 2020

Keywords:
Nuclear power plant
Procedure
Information extraction
Natural language processing
Rule
Ontology

ABSTRACT

Procedures play a key role in ensuring safe operation at nuclear power plants (NPPs). Development and maintenance of a large number of procedures reflecting the best knowledge available in all relevant areas is a complex job. This paper introduces a newly developed methodology and the implemented software, called *iExtractor*, for the extraction of syntactic and semantic information from NPP procedures utilizing natural language processing (NLP)-based technologies. The steps of the *iExtractor* integrated with sets of rules and an ontology for NPPs are described in detail with examples. Case study results of the *iExtractor* applied to selected procedures of a U.S. commercial NPP are also introduced. It is shown that the *iExtractor* can provide overall comprehension of the analyzed procedures and indicate parts of procedures that need improvement. The rich information extracted from procedures could be further utilized as a basis for their enhanced management.

1. Introduction

Procedures play a key role in ensuring safe, deliberate, and controlled operations at a facility by broadly supporting the activities of its personnel [1–5]. They provide the interface between the equipment and the personnel who operate and maintain them [1]. They also play an intermediary role in the transfer of knowledge from system engineers to the operators of the system and even for training purposes [5]. Furthermore, relevant procedures support the plant managers' understanding of how exactly to meet the standards and expectations for the operation and maintenance of the plant [1].

Procedures must be technically and operationally accurate integrating the up-to-date knowledge available in all relevant areas, which include the requirements, policies, physical facilities, processes, and people necessary to operate the facility safely [2,3]. In addition, the controlled documents of procedures must be easy to follow to ensure human performance quality by clearly providing the purpose, specific intent, and sequenced direction for an activity, program, or process [2]. A facility needs a large quantity of cross-referenced procedures, depending on the plant scale and the complexity of its processes. Ensuring that this large volume of procedures meets the above criteria is a massive job [3].

Faced with increasingly competitive energy markets, it is crucial for a facility to be operated and maintained in more efficient and effective ways [2]. For such purposes, NPPs are employing more advanced systems integrated with the digital technologies [2,6–8]. Innovative solutions are sought-after to ensure the development of sound procedures and their continuous improvement in more efficient manner [1]. This paper introduces a newly developed methodology and the implemented software, called *iExtractor*, which automatically captures the syntactic and semantic information from NPP procedures as an essential tool for the enhanced management of procedures.

The rest of this paper is organized as follows: Section 2 briefly introduces the requirements of procedures and the state-of-the-art technologies of extracting needed information from texts. Section 3 describes the features of the newly developed methodology and the software *iExtractor* in detail with examples. Section 4 introduces the findings from the case study results. Finally, section 5 gives the conclusions and suggests future research directions.

[*] Corresponding author.
E-mail address: yschoi@inje.edu (Y. Choi).

https://doi.org/10.1016/j.net.2020.08.010

2. Backgrounds and preliminaries

2.1. Broad requirements for managing procedure programs

As depicted in Fig. 1, the IAEA-TECDOC-1651 placed procedures as a major component of the configuration information program [9]. It emphasized that procedures need to be consistent with other components in a timely manner to ensure that safe, technically sound, and cost-effective decisions are made with confidence. The IAEA-TECDOC-1058 aimed to provide good practices with respect to the development and the use of NPP procedures, based on the historic lessons learned from NPPs and utilities [1]. More specifically, it presented foundational issues of procedure system development methodology.

The U.S. Department of Energy (DOE) also emphasized the role of procedures as a key component for the overall safe operation of a facility. The DOE-STD-1029-92, Writer's Guide for Technical Procedures [3], defined the broad requirements for managing procedure programs and provided guidance on the process of developing and maintaining the procedures at DOE facilities. It introduced detailed issues regarding the processes involved in writing technical procedures, which include establishing the basis, content and format of a procedure, and writing and structuring the action steps.

The Procedure Professionals Association (PPA) has developed voluntary consensus standards, AP-907-005, Procedure Writers' Manual [2], in conjunction with AP-907-001, Procedure Process Description [10]. The U.S. DOE concluded that those PPA documents adequately fulfilled the purpose of DOE-STD-1029-92 and endorsed the PPA as suitable for further work on those issues [2]. The AP-907-005 provides a nuclear industry consensus standard for writing human-factored procedures with elaborately developed specific guidelines. These guidelines are considered the de facto standards by many U.S. commercial NPPs and other industrial plants worldwide.

A large number of technical procedures are needed for diverse NPP components, requiring large number of people in the development and improvement of procedures. Their technical backgrounds and work experiences with specific components are diverse, and their understandings of the requirements of procedures are not necessarily the same. Thus, coherent review of applicable operating experiences, outstanding issues, human performance challenges as well as technical contents of procedures is a challenging task [10]. This study was motivated by a desire to reduce the burden of these tasks and to make the process more efficient for the development and maintenance of sound and effective procedures.

2.2. NLP-based information extraction integrated with rules and domain ontology

Information extraction is the process by which data from machine-readable documents is selectively structured and combined [11]. In general, natural language processing (NLP) technique is employed to analyze the texts in the input documents before extracting their information. Information extraction enables much richer forms of queries on the abundant unstructured sources than possible with keyword searches alone [12].

Information extraction approaches are classified into two main types: knowledge engineering and automatic training [13]. Knowledge engineering (KE) approach utilizes the domain knowledge of human expertise represented in a machine-understandable form. The domain knowledge is often represented in the form of production rules, mostly in the form of Common Pattern Specification Language (CPSL) grammar [14] or its derivatives, like Java Annotation Patterns Engine (JAPE) grammar [15]. Each pattern/action rule consists of conditional patterns and action statements for annotation. Thus, the KE approach is also referred to as a rule-based approach. Rules are iteratively constructed and refined to improve the accuracy of text processing [16]. The automatic training approach, also known as the machine learning (ML) approach, utilizes ML algorithms. In general, the ML approach requires a large amount of annotated training data, on which its performance is dependent, and often results in inconsistent and insufficient outcomes [17,18]. The KE approach tends to yield higher performance, which is explained by the assertion that human expertise often results in more accurate patterns and extraction rules [18]. The efforts required for defining patterns and developing rules in the KE approach are expected to be less than those required for manually annotating a sufficiently large size of training data in the ML approach [18].

A domain-specific information extraction results in more suitable outcomes than when applied to general non-technical texts due to the reduction in homonym conflicts and co-reference resolution problems and the enhanced interpretability of domain-specific texts [16]. Domain-specific information extraction methodologies are often integrated with domain ontologies to enhance their performances [16,18,19]. An ontology is a specification of knowledge in a certain domain. It includes machine-interpretable definitions of concepts in the domain and relations among them,

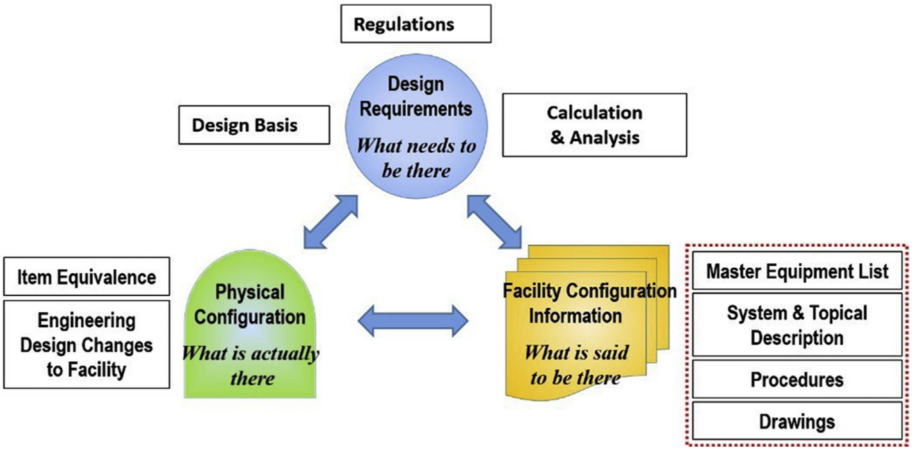


Fig. 1. Configuration equilibrium diagram with related programs [9].

where each concept is associated with a set of its own instance entities [20]. Ontology building, also known as ontology learning, is approached in several ways depending on the techniques used, including statistics, linguistics, machine learning and logical inference, and hybrid methods. A detailed review of such approaches can be found in Ref. [19].

There have been applications of the NLP techniques in the nuclear industry, for checking conformance to requirement templates [21], identifying causal relationships from event reports [22], answering natural language search queries in the nuclear domain [23] utilizing Google BERT [24], etc. Applications of ontologies could be found in the nuclear industry [25–27] as well as in other industries [28–30]. Fig. 2(a) shows a part of the hierarchical representation of NPP components [31] utilizing Protégé [32]; Fig. 2(b) illustrates relationships among major NPP components [25].

3. The methodology and the software tool *iExtractor*

The *iExtractor* automatically captures the syntactic and semantic information from the input procedures in three phases: preprocessing, natural language processing, and the main phase of information extraction. All the three phases are implemented in Microsoft C#, employing some publicly available software components at the first two phases. Compared with related systems, like the ANNIE system of GATE [33] and Stanford NER [34], the *iExtractor* has unique features to extract more valid information from procedures. These include enhancement of POS tagging, classification of paragraph types, and identification of step statement components, each integrated with its own rule set. This section briefly describes the steps of the *iExtractor*.

3.1. Pre-processing phase

The *iExtractor* system preprocesses the input procedure as follows utilizing the API tools provided by the word processor software, e.g., the Microsoft Word Reader API [35]:

1) For each image object (figures, drawings, charts, etc.), its location is marked and stored separately for further processing.
2) For each table, the structural information including the nesting relation is noted and any image object in the table is handled in the same manner as above.
3) For each text paragraph (including those in tables), which is a series of texts ending with a hard return by the Enter key, its structural properties (text index, bullet offset, etc.) and rich text features (typeface, size, color, underline, etc.) are extracted. Additional structural properties (parent paragraph, texts at margins, index level, etc.) are also extracted. Each text paragraph is converted into a new instance of the ‘paragraph’ class, which is the main data structure of the *iExtractor.* This data structure will be enriched by further storing all the information extracted at each step of the proposed method to its corresponding slots.

3.2. Natural language processing phase

The *iExtractor* system utilizes publicly available natural language processing tools in accordance with the language used to describe the procedures, such as the Stanford CoreNLP Toolkit [36] or Open Korean Text [37], which support the following sub-steps, in general. Fig. 7, at the end of section 3, illustrates the stepwise outcomes of the selected steps in sections 3.2 and 3.3, for the target statement shown in its first row.

3.2.1. Tokenization, sentence splitting and lemmatization

The NLP tool first splits each text paragraph into token(s) of texts and then into sentence(s). If the word contained in a token is in inflectional or derivational form, then the root form of the word is also provided to the token. More detailed descriptions for the three sub-steps follow:

1) Tokenization splits texts into separate tokens, each containing any type of text such as a word, a number, a punctuation mark, a symbol, etc. [38]. A token could be associated with additional attributes for the text it contains, such as its length, the start and the end position indices. The second row of Fig. 7 illustrates tokenization results for the statement shown in the first row of Fig. 7.

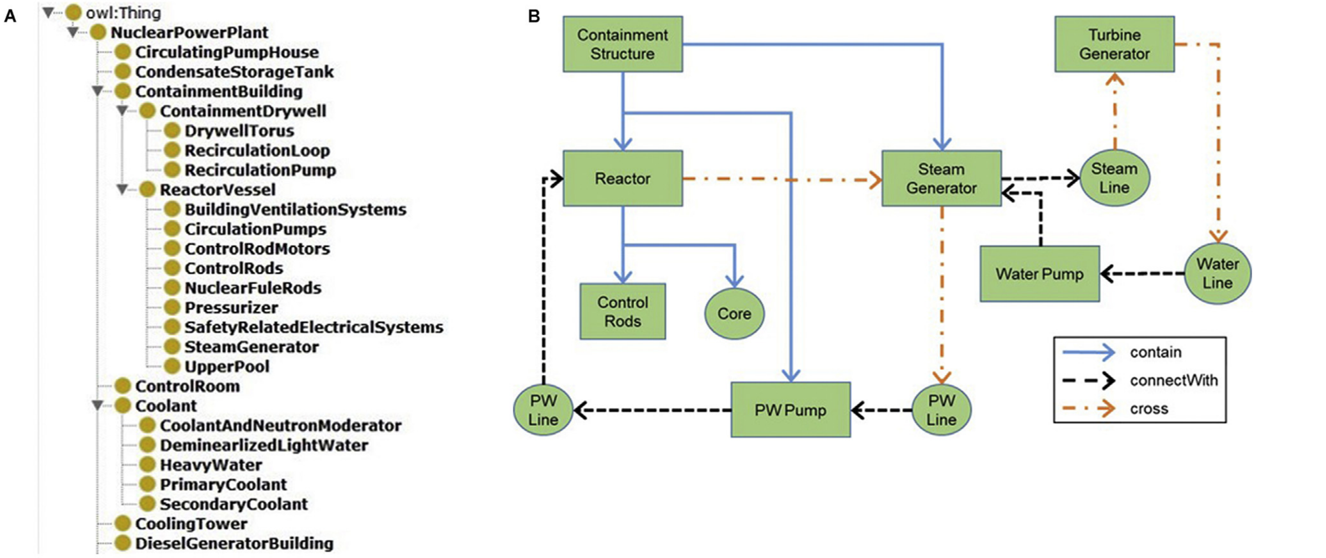


Fig. 2. Examples of ontologies in the nuclear industry. (a) Hierarchical representation of NPP components (in part) [31]; (b) Simplified relationships among major NPP components [25].

2) Sentence splitting splits the text paragraphs into separate sentences utilizing the sentence boundary indicators such as periods, question marks, exclamation points [39].

3) Lemmatization utilizes morphological analysis to derive the root form of each word in inflectional form or in derivational form [39]. Lemmatization helps to look up the instances of concepts, not represented in the root form, in the input procedure.

3.2.2. Part-of-speech (POS) tagging and parsing

POS tagging is the task of labeling each token with its part-of-speech such as noun, verb, adverb. Parsing is the process of constructing a hierarchical structure tree of tokens for each sentence [38]. The third row of Fig. 7 illustrates the POS tags for the tokens represented in the second row. Table A1 in Appendix A provides descriptions of the tags and relation labels used in this paper, in alphabetical order, and their full lists are provided in Ref. [40,41].

There are two forms of parse trees, the constituency-based and the dependency-based. A constituency-based parse tree shows the hierarchical structure of the constituents, each as a group of token(s), by the phrase structure grammar [42]. Fig. 3(a), for instance, shows the constituency-based parse tree for the statement shown in the first row of Fig. 7. In Fig. 3(a), each leaf node is the POS tag for a token, with its text represented at the bottom, whereas each non-leaf node is a constituent tag. A dependency-based parse tree is a directed tree with nodes of lexical tokens where each directed edge represents the grammatical relation between a pair of nodes [42]. Fig. 3(b) shows the dependency-based parse tree for the same statement in Fig. 3(a). In Fig. 3(b), each label on an edge indicates the grammatical role of the child token to the parent token, described in Table A1 in Appendix A.

3.2.3. POS tagging enhancement

Neither the POS tag of a word nor the parse tree of a sentence is unique. Existing tools recommend the most probable one, based on some metric, from multiple possible choices for each word or sentence, respectively [38]. Thus, for general unstructured sentences, without any annotated reference corpus, a tool-driven evaluation for the POS tags and the parse trees is meaningless [43]. Any evaluation and correction of a tag for a token and/or a parse tree for a sentence requires manual work [40].

For procedures, however, many domain-specific terminologies are utilized. In addition, each step statement is represented in a semi-structured form, mostly in an imperative form and possibly preceded by additional components. Some words of syntactic or semantic importance, such as conditional or logic terms, action verbs, or codes, are capitalized as a way of emphasis purpose. Those special features of procedures often limit the performance of POS tagging and parsing, and further the performance of the final information extraction [22].

The *iExtractor* system evaluates the POS tags for tokens utilizing a customized lexical database for NPPs, which is a miniature version of Wordnet [44], and the special features of procedure statements. Misinterpreted POS tags are detected and corrected by simple built-in rules integrated with the lexical database.

Fig. 4, below, shows the initial parse tree for a statement of:

‘**IF** as-found thickness is less than 0.180 inches,
THEN REPLACE X Relay Lever per 0-MNT-005, Relay Replacement.’

It is shown that the tokens of ‘**THEN**’ and ‘**REPLACE**’ are wrongly tagged as ‘NNP’ (proper noun, singular [40]) and ‘VBP’(verb, non-3rd person singular present [40]), respectively. Consequently, ‘THEN’ is also wrongly interpreted as the nominative of the verb ‘REPLACE’, that is its constituent tag is wrongly assessed as ‘NP’(noun phrase [40]).

Fig. 5 shows the enhanced parse tree after applying one of the POS tagging enhancement rules associated with the ‘**IF** < *condition(s)*>, **THEN** <*action(s)*>’ statement type, in which the tag for the token of ‘**THEN**’ is changed into ‘RB’(adverb [40]). This rule-based POS tag correction for a token leads to additional changes of tags for other tokens and constituents by re-parsing the statement. The POS tag for the token of ‘REPLACE’ is changed into ‘VB’ (verb, base form [40]) and the constituent tag of ‘THEN’ is changed into ‘ADVP’ (adverb phrase [40]), which are now in accordance with what the statement implies.

3.3. Information extraction phase

Utilizing the outcomes of the first two phases, three types of information are extracted for each paragraph via corresponding steps: semantic entities, paragraph types, and step statement components. The last type of extraction is used only for applicable paragraph types, to be described in section 3.3.3.

3.3.1. Semantic entity tagging

This step annotates selected words in each paragraph with their semantic type in two sub-steps: first, by ontology lookup and then, by built-in rules. Through the ontology lookup, any token(s) containing word(s) matched with an instance entity in the ontology are tagged as the corresponding concept. The fourth row of Fig. 7

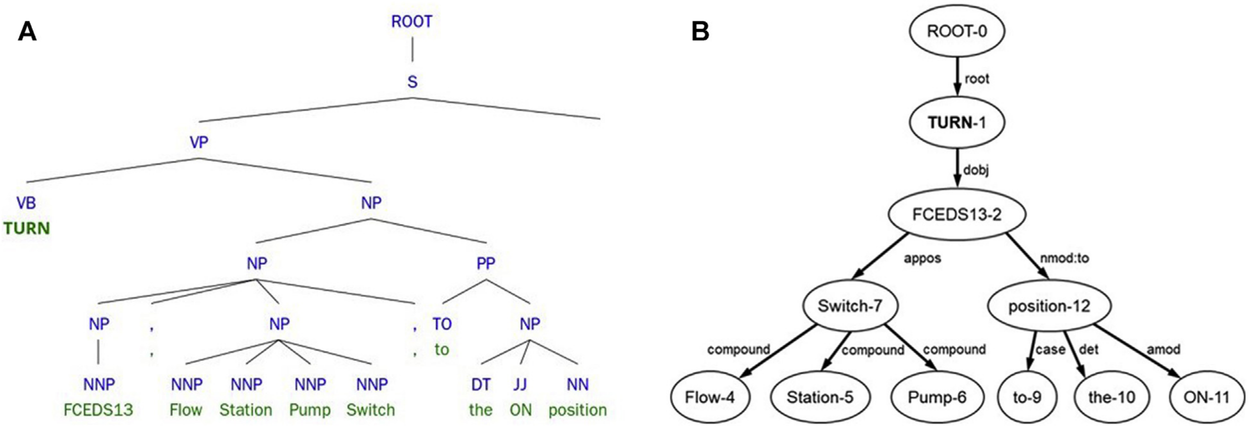


Fig. 3. Examples of two types of parse trees. (a) Constituency-based parse tree; (b) Dependency-based parse tree.

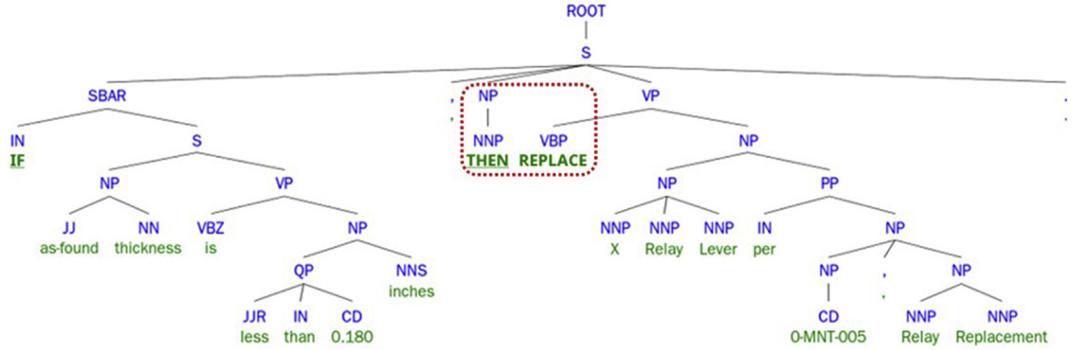


Fig. 4. Initial parse tree with improperly assessed tags.

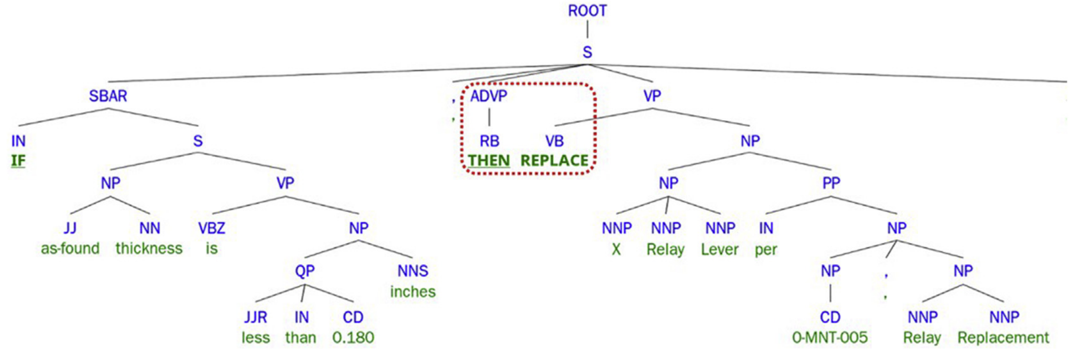


Fig. 5. The enhanced parse tree after applying rule-based POS tagging correction.

illustrates the lookup tags (or annotations) for the tokens represented in the second row.

The following lists of concepts were introduced to the ontology in association with the three questions that need to be answered for action step statements [2]:

- *WHO performs the specific task?* - organization, division, role, staff;
- *WHAT task is to be performed?* - action verb, structure, system, component, and part (SSCP);
- *HOW to correctly perform the task in a safe and efficient manner?* - tool, material, measure, measure unit, criteria, status.

Additional lists of concepts were also introduced as those frequently appear in actual procedures, such as documents and codes. The initial set of instance entities of each concept, both in English and in Korean, was developed utilizing multiple sources, such as [2,45,46], and the actual procedures. The sets of concepts and lists of instance entities of each concept have passed through several iterations of improvement, in cooperation with domain experts [19], reflecting newly collected evidences [16,19].

Action verbs are key elements of step statements that describe directives to be performed by the procedure user. They are identified, not by simple ontology lookup, but as verbs satisfying all of the following conditions:

- A verb in base form without conjunction with ‘to’;
- A verb neither associated with a nominative nor in conjunction with another verb having a nominative;
- In the case of the verb ‘have’, it is not involved with any tense such as present perfect tense.

Lookup-based tagging annotates only the tokens matched with the semantic entities included in the ontology. However, an ontology contains only some limited set of representative entities for a certain problem area in a specific domain. The *iExtractor* system utilizes the rule-based semantic annotation, where any token(s) satisfying the condition(s) of a rule is/are tagged with the concept designated by that rule [33]. Both syntactic and semantic features of texts were employed in the rules. For semantic entities missed by the ontology lookup and by the current rule set, the ontology is expanded with their representative entities [16,19] or those procedure statement patterns are transformed into new pattern/action rules [14,15]. For wrongly tagged semantic entities, the corresponding rules are refined [16].

As an example, the following rule, represented by CPSL [14] grammar, designates a new SSCP instance as the consecutive tokens of nouns plus a succeeding SSCP instance found by the ontology lookup.

```
((({POS = NN}|{POS = NNP})*
    {OntologyLookup = SSCP}): SSCPinstance
    {POS !=~ "NN.*"})

     → SSCP =: SSCPinstance
```

The fifth row of Fig. 7 illustrates a new semantic tag annotated by the above rule. The tokens of ‘Pump Switch’, tagged as an SSCP entity by the ontology lookup, are preceded by two consecutive singular proper nouns, each POS tagged as ‘NNP’ as shown in the third row of Fig. 7. Then, the above rule annotates this compound noun of four tokens of words, ‘Flow Station Pump Switch’, as a new SSCP entity represented in the fifth row of Fig. 7.

3.3.2. Classification of paragraph types

The *iExtractor* also determines the type of each paragraph, which specifically indicates how it is to be further processed. Brief descriptions of each paragraph type, classified into three groups, are provided below.

First, each paragraph that contains *action verb(s)* and *target object(s)* is classified into the *step* group. Specific types of paragraphs belonging to this group, based on the syntactic or semantic components shown in Fig. 6, include the following:

- A (non-conditional) *action step* typically starts with an *action verb*, possibly preceded by additional component(s) of *critical information* and/or an *adverb.*
- A *branching step* (**GO TO** ~ or **PROCEED TO** ~) or a *referencing step* (**REFER TO** ~, **SEE** ~, **USE** ~, **REPEAT** ~, or **PER**) is a special form of step that starts with those designated action verbs.
- A *conditional action step* (**IF/WHEN** < *condition(s)*>, **THEN** <*action(s)*>) or a *continuous action step* (**WHILE/IF AT ANY TIME** < *condition(s)*>, <*action(s)*>), is a special form of a step which starts with those designated keywords. The *condition(s)* could be a compound of multiple conditional clauses, each connected with another one by a logical operator, e.g., **AND**, or a list of conditional clauses after the clause including like ‘any of the following’. The *action(s)* clause is usually in a form of an action step.

The second group of paragraphs includes the following that are closely related with specific action step(s):

- Each statement following **NOTE**, **CAUTION**, or **WARNING** (each to be classified as a *NCW header*) is to be classified as an *NCW statement.* Each of those is placed prior to relevant step(s) and shall not contain any directive [2].
- The paragraph of **HOLD POINT** or in the form preceded with a topical keyword, like **QA HOLD POINT**, is classified as a *hold point* type.
- In some steps, it is required to record the observed data and further the calculated values(s) utilizing a simple formula involving newly observed data. Such paragraphs, placed after relevant step statements, are classified as a *record row* or a *calculation row*, respectively. A signoff could be represented as an independent paragraph to be classified as a *signoff row.*
- In case any logical operator itself builds a paragraph, it is classified as a *logical operator.* A *list element* is each paragraph in bullet form listed typically after the step statement containing ‘the following’.

The third group of paragraphs are not intrinsically related to specific action step(s):

- The *(sub)section title and* the *caption* of a figure or a table.
- The *continuation heading* which denotes a page break that a step continues onto another page.
- The final *information* type is for paragraphs that are not any of the above types.

3.3.3. Identification of step paragraph components

According to PPA [2], a step statement is decomposed into six components, as shown in Fig. 6. Two mandatory core components, the action verb(s) and the target object(s), and four optional ones in brackets. These components are all identified by the *iExtractor* system from any paragraph of those five types belonging to the first step group.

For this purpose, this step also utilizes the built-in rules associated with the prior extracted tags for tokens. The conditions of each rule are expressed with POS tags, constituent tags, and relation tags of tokens. For each paragraph, the *iExtractor* finds the rule matching the current paragraph and identifies the components designated by that rule. As an example, the following rule identifies three components of action verb, target object, and supporting information from a paragraph of action step type matched with all three conditions.

```
(({Dependency = dobj} {SemanticType = ActionVerb}):av
       {Node = NP | POS = NN | POS = CD}:to1)
 ({Dependency = appos} {Node = NP | POS = NN | POS = CD}:to1
       {Node = NP | POS = NN | POS = CD}:to2)
 ({Dependency = nmod} {Node = NP | POS = NN | POS = CD}:to1
       {Node = PP | Node = VP | Node = SBAR}:si))
  → Action verb =: av, Target object =: to1 + to2,
    Supplemental information =: si.
```

The sixth row of Fig. 7 illustrates the three statement components identified by the above rule. From Fig. 3(a) and (b), it can be deduced that the three conditions of the above rule hold, with four arguments of **‘TURN’** as `av`, ‘FCEDS13’ as `to1`, ‘Flow Station Pump Switch’ as `to2`, and ‘to the ON position’ as `si`. Thus, the above rule designates the three components, from the left to the right respectively as `av`, `to1` + `to2`, and `si`, as shown in the sixth row of Fig. 7.

All the information extraction results are stored into a database and are produced in various additional forms by the *iExtractor* system. Fig. 8 shows two examples of those: (a) the procedure document with semantic entities highlighted in different colors according to their concepts; and (b) the procedure document with paragraphs highlighted in different colors according to their types. Procedures are important assets of a specific facility and are not allowed to be disclosed. Thus, the color-highlighted procedures in Fig. 8 are only for illustration, applied to a sample procedure introduced in Refs. [47]. Various outputs provided by the *iExtractor* system, including those in Fig. 8, are also utilized to collect feedback on the information extraction results from domain experts and to improve its performance by extending the ontology and the rules.

4. Case study results

The methodology and the software *iExtractor* introduced in section 3 have been applied to twenty-five procedures obtained from a U.S. commercial NPP in two groups: ten operating procedures (OPs) and fifteen testing procedures (TPs). This section briefly introduces selected findings associated with the relevant summary statistics obtained by simple queries to the database of all the information extracted from those procedures.

This experimental analysis focused on the *instructions* sections, where each contained step statements. The (sub)section of *prerequisites* (for both OPs and TPs) and the (sub)sections of *restoration* and *acceptance verification* (for TPs only) were handled as parts of the *instructions* section. The number of analyzed procedures was

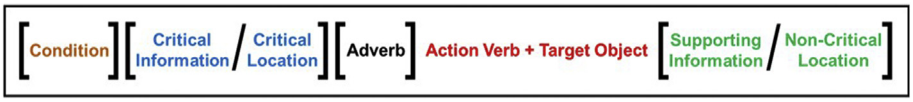
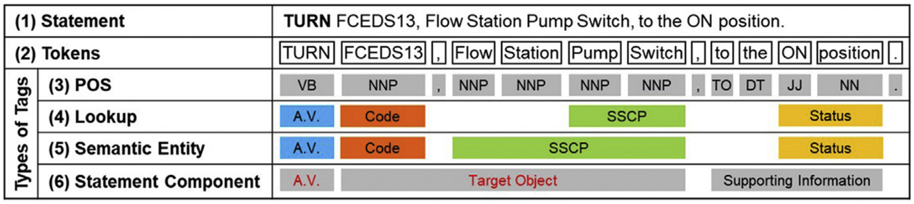
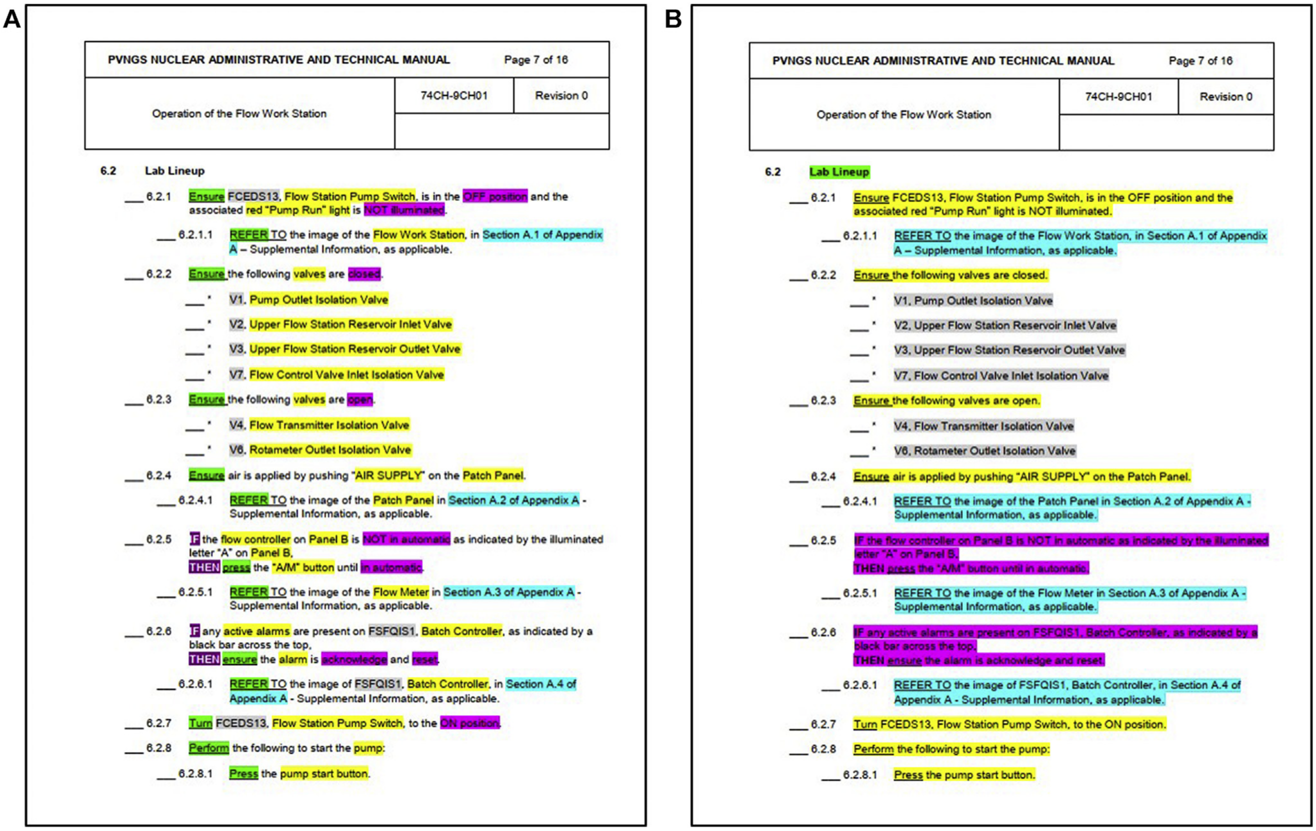


Fig. 6. Syntactic and semantic components in step statements [4].



(1) Statement		TURN FCEDS13, Flow Station Pump Switch, to the ON position.												
(2) Tokens		TURN	FCEDS13	,	Flow	Station	Pump	Switch	,	to	the	ON	position	.
Types of Tags	(3) POS	VB	NNP	,	NNP	NNP	NNP	NNP	,	TO	DT	JJ	NN	.
	(4) Lookup	A.V.	Code				SSCP					Status		
	(5) Semantic Entity	A.V.	Code		SSCP							Status		
	(6) Statement Component	A.V.	Target Object							Supporting Information				

Fig. 7. An example statement and the results of selected steps in sections 3.2 and 3.3.



PVNGS NUCLEAR ADMINISTRATIVE AND TECHNICAL MANUAL	Page 7 of 16	
Operation of the Flow Work Station	74CH-9CH01	Revision 0

6.2 Lab Lineup

___ 6.2.1 Ensure FCEDS13, Flow Station Pump Switch, is in the OFF position and the associated red "Pump Run" light is NOT illuminated.

___ 6.2.1.1 REFER TO the image of the Flow Work Station, in Section A.1 of Appendix A – Supplemental Information, as applicable.

___ 6.2.2 Ensure the following valves are closed.

___ • V1. Pump Outlet Isolation Valve

___ • V2. Upper Flow Station Reservoir Inlet Valve

___ • V3. Upper Flow Station Reservoir Outlet Valve

___ • V7. Flow Control Valve Inlet Isolation Valve

___ 6.2.3 Ensure the following valves are open.

___ • V4. Flow Transmitter Isolation Valve

___ • V6. Rotameter Outlet Isolation Valve

___ 6.2.4 Ensure air is applied by pushing "AIR SUPPLY" on the Patch Panel.

___ 6.2.4.1 REFER TO the image of the Patch Panel in Section A.2 of Appendix A - Supplemental Information, as applicable.

___ 6.2.5 IF the flow controller on Panel B is NOT in automatic as indicated by the illuminated letter "A" on Panel B,
THEN press the "A/M" button until in automatic.

___ 6.2.5.1 REFER TO the image of the Flow Meter in Section A.3 of Appendix A - Supplemental Information, as applicable.

___ 6.2.6 IF any active alarms are present on FSFQIS1, Batch Controller, as indicated by a black bar across the top,
THEN ensure the alarm is acknowledge and reset.

___ 6.2.6.1 REFER TO the image of FSFQIS1, Batch Controller, in Section A.4 of Appendix A - Supplemental Information, as applicable.

___ 6.2.7 Turn FCEDS13, Flow Station Pump Switch, to the ON position.

___ 6.2.8 Perform the following to start the pump:

___ 6.2.8.1 Press the pump start button.

PVNGS NUCLEAR ADMINISTRATIVE AND TECHNICAL MANUAL Page 7 of 16
Operation of the Flow Work Station 74CH-9CH01 Revision 0

6.2 Lab Lineup

___ 6.2.1 Ensure FCEDS13, Flow Station Pump Switch, is in the OFF position and the associated red "Pump Run" light is NOT illuminated.

___ 6.2.1.1 REFER TO the image of the Flow Work Station, in Section A.1 of Appendix A – Supplemental Information, as applicable.

___ 6.2.2 Ensure the following valves are closed.

___ • V1. Pump Outlet Isolation Valve

___ • V2. Upper Flow Station Reservoir Inlet Valve

___ • V3. Upper Flow Station Reservoir Outlet Valve

___ • V7. Flow Control Valve Inlet Isolation Valve

___ 6.2.3 Ensure the following valves are open.

___ • V4. Flow Transmitter Isolation Valve

___ • V6. Rotameter Outlet Isolation Valve

___ 6.2.4 Ensure air is applied by pushing "AIR SUPPLY" on the Patch Panel.

___ 6.2.4.1 REFER TO the image of the Patch Panel in Section A.2 of Appendix A - Supplemental Information, as applicable.

___ 6.2.5 IF the flow controller on Panel B is NOT in automatic as indicated by the illuminated letter "A" on Panel B,
THEN press the "A/M" button until in automatic.

___ 6.2.5.1 REFER TO the image of the Flow Meter in Section A.3 of Appendix A - Supplemental Information, as applicable.

___ 6.2.6 IF any active alarms are present on FSFQIS1, Batch Controller, as indicated by a black bar across the top,
THEN ensure the alarm is acknowledge and reset.

___ 6.2.6.1 REFER TO the image of FSFQIS1, Batch Controller, in Section A.4 of Appendix A - Supplemental Information, as applicable.

___ 6.2.7 Turn FCEDS13, Flow Station Pump Switch, to the ON position.

___ 6.2.8 Perform the following to start the pump:

___ 6.2.8.1 Press the pump start button.

Fig. 8. Two types of highlighting applied to a sample page of a procedure in [47]. (a) Semantic entities; (b) Paragraph types.

not sufficient to deduce some statistically significant conclusions. Moreover, the verification of procedures' compliance to writing guidelines was not the main target of this experimental analysis. Considering the facts, however, it is apparent that the *iExtractor* can provide overall comprehension of the analyzed procedures and indicate which parts of procedures need to be improved. Recommendations on how to improve those are also provided.

Table 1 shows the statistics regarding the number of pages of the procedures analyzed, of two specific sections and of the whole procedure. All the values in Table 1 were computed row-wise, thus column-wise interpretation is meaningless. On average, it shows that OPs have more pages than TPs for the *instructions* section (76.2 vs. 19.7) as well as for the whole procedure (129.6 vs. 37.7). The *instructions* section takes up, on average, 55.8% of the total for OPs and 49.4% for TPs. It is shown that the *attachment* section also takes up a sizable number of pages in those procedures, 31.0% of the total for OPs and 12.7% for TPs.

Table 2 shows the counts of non-text objects (figures, tables) in procedures, where tables are classified into four types: a plain table that provides relevant information in structured form; and each of the other three types of tables that have designated space(s) for placekeeping, to record observed data, and to record value(s) calculated from newly observed data. It shows that many of those non-text objects appear in the *attachment* section, especially with OPs.

Table 1
Statistics on the number of pages for each procedure.

	OPs			TPs		
	Min	Max	Avg.	Min	Max	Avg.
Instructions section (A)	25	180	76.2	5	46	19.7
Attachment section (B)	7	87	38.1	2	36	5.3
Whole procedure (C)	39	232	129.6	18	110	37.7
A over C (%)	25.0	79.2	55.8	26.9	74.6	49.4
B over C (%)	8.3	63.0	31.0	3.4	34.6	12.7

Table 2
Incidence counts of non-text objects.

Type of non-text objects		OPs			TPs		
		A	B	Others	A	B	Others
Figure		–	57	–	–	6	–
Table	Plain	5	300	30	–	30	54
	Placekeeping	–	–	–	–	–	–
	Record	1	–		–	–	
	Calculation	–	–	–	9	–	

(A: Instructions section, B: Attachment section, as in Table 1.)

4.1. Semantic annotation results

This section will focus on the semantic annotation results for action verbs. Table 3 shows the numbers of different action verbs employed in each group of procedures. Table 4 shows the values, the minimum, the maximum, and the average, of the different action verbs used for each procedure in each group. From the tables, it can be seen that OPs utilized relatively more distinct action verbs than TPs, which might be because OPs have relatively more pages than TPs or that OPs technically require more distinct actions than TPs. It can also be deduced that many action verbs were utilized over multiple procedures. It is additionally shown that this plant utilized many facility-specific action verbs other than those listed in Ref. [2]; however, their incidence counts are slightly lower than those of PPA-listed ones.

Table 5 shows the top five action verbs used for each procedure group and by the type of action verbs, PPA-listed ones or facility-specific ones.

- Four of the top five PPA-listed action verbs overlap for both OPs and TPs, and they are again included in the 'All' procedures column.
- For facility-specific action verbs, as expected, it shows that their incidence counts were much lower than those of the PPA-listed ones. Interestingly, there is no overlap between the two top five facility-specific action verbs for OPs and for TPs.

It is encouraged to use facility-specific action verbs, however, each action verb needs to be clearly defined with its necessity and shared over the entire facility [2]. The full list of action verbs with each incidence count, provided by the *iExtractor*, helps to refine the set of action verbs. Reviewed together with their definitions, some action verbs (especially those with very low counts) could be replaced with other ones and screened out as dispensable.

Table 3
Number of different action verbs.

	OPs	%	TPs	%	All	%
PPA-listed	127	61.4	67	67.7	132	58.9
Facility-specific	80	38.6	32	32.3	92	41.1
Total	207	100.0	99	100.0	224	100.0

Table 4
Statistics of the number of different action verbs in each procedure.

	OPs			TPs		
	Min	Max	Avg.	Min	Max	Avg.
PPA-listed	26	66	44.4	9	42	20.0
Facility-specific	8	22	15.0	3	14	5.7

As shown in Table 5, the action verbs in the form of 'Do not (verb)' occur thirty times in OPs. Actually, there is one more occurrence with a TP, which is not shown in Table 5. It is contrary to the procedure writing guideline to write steps as positive statements [2]. Table 6 shows the types of paragraphs containing action verbs in the form of 'Do not (verb)'. Three of those paragraphs are of the NCW statement type, two caution statements and one warning statement. The other 28 paragraphs belong to the step group, with or without the conditional (or continuous) component. NCW statements are prohibited to contain any directive, thus, it is recommended to revise the three paragraphs not in the form of a step statement. And, it is also recommended to revise the other 28 paragraphs in positive statements each with the same context.

4.2. Paragraph type classification results

4.2.1. Incidence of each paragraph type

Table 7 shows the incidence count of each paragraph type in each procedure group. The following were found for the first group of paragraphs:

- The five types of step paragraphs made up slightly less than half of the total counts both for OPs (48.37%) and for TPs (47.22%).
- The action step (37.10% for OPs and 41.28% for TPs) and the conditional action step (10.17% for OPs and 5.51% for TPs) were the dominant types in this group of paragraphs.
- Paragraphs of the continuous action step type were used to some extent with OPs, but rare with TPs.

For the second group of paragraphs, the following were found:

- Paragraphs of the list element type were used in large numbers for both OPs and TPs.
- A fair number of paragraphs of the NCW statement type were found, more with OPs.
- No paragraph of the hold point type was found for either OPs or TPs; Paragraphs of the calculation row type were rarely used for both OPs and TPs.
- Paragraphs of the record row type and the paragraphs of the signoff row type were used to some extent with TPs, but much less with OPs.

For the third group of paragraphs, the following were found:

- No captions were found for six tables in OPs and nine tables in TPs, for those contained in the instructions sections as shown in Table 2.
- Paragraphs of the information type were found to some extent for both OPs and TPs, which need careful reviews, to be described in section 4.2.4.

Table 5
Top five frequently used action verbs.

	OPs		TPs		All	
	Action verb	Count	Action verb	Count	Action verb	Count
PPA-listed	Ensure	933	Verify	522	Verify	1095
	Verify	573	Perform	172	Ensure	1056
	Open	467	*Record*	154	Perform	594
	Perform	422	Ensure	123	Open	565
	Close	407	Open	98	*Close*	490
Facility-specific	Crack	31	Sign	63	Sign	63
	Do not (verb)	30	Review	27	Review	37
	Rotate	30	Unbypass	20	Have	33
	Wait	27	Time	19	Do not (verb)	31
	Note	26	Observe	10	Crack	31

Table 6
Detailed types of paragraphs containing action verbs in the form of 'Do not (verb)'.

Paragraph type	OPs	TPs
Action Step	17	1
Conditional Action Step	7	–
Continuous Action Step	3	–
Caution Statement	2	–
Warning Statement	1	–
Total	30	1

4.2.2. Additional information at the left or right margins of paragraphs

Not as an independent paragraph, texts at the left or right margins of paragraphs are frequently utilized for various purposes in procedures, mostly for placekeeping purposes. Placekeepings are utilized to help procedure users track performance of steps within a procedure by physically marking steps that have been completed or are not applicable [2]. In general, underlined white spaces, checkboxes, or signature lines are used for placekeeping. Table 8 shows the incidence counts of paragraphs each accompanied with any type of placekeeping for each paragraph type. It was found that all placekeepings are placed at the right margin and employed only for TPs. To save space, only the rows of paragraph types containing placekeeping are shown. From Table 8, it can be seen that placekeeping is employed quite often for the paragraphs of the step group and the list element type.

4.2.3. More on the paragraphs of the types of conditional or continuous action step

Table 9 shows more detailed classification of paragraphs belonging to the types of conditional action step or continuous action step. For a conditional action step, it is required to place **THEN** between *condition(s)* and *action(s)*. But, not necessarily for a continuous action step [2].

- For paragraphs of the conditional action step type starting with **IF**, most of those followed this requirement; however, 17 paragraphs (3 in OPs and 14 in TPs) violated that requirement. For paragraphs of the conditional action step type starting with **WHEN**, on the other hand, only 6 of those with OPs followed this requirement and all of the other 317 paragraphs (275 with OPs and 42 with TPs) violated the requirement. For each of those violating paragraphs, it is recommended to insert **THEN** right before the action(s) clause.

Table 7
Incidence count for each paragraph type.

Types of paragraph		OPs		TPs	
		Count	%	Count	%
Step	Action step	4022	37.10	1731	41.28
	Branching (step)	2	0.02	0	–
	Referencing (step)	22	0.20	12	0.29
	Conditional action step	1103	10.17	231	5.51
	Continuous action step	95	0.88	6	0.14
	Subtotal	*5244*	*48.37*	*1980*	*47.22*
Related with steps	NCW header	830	7.66	174	4.15
	NCW statement	860	7.93	174	4.15
	Hold point	0	–	0	–
	Record row	11	0.10	159	3.79
	Calculation row	3	0.03	3	0.07
	Signoff row	1	0.01	214	5.10
	List element	3239	29.87	1060	25.28
	Logical operator	70	0.65	17	0.41
	Subtotal	*5014*	*46.25*	*1801*	*42.95*
Others	(sub)Section title	391	3.61	182	4.34
	Figure/Table caption	0	–	0	–
	Continuation heading	103	0.95	50	1.19
	Information	90	0.83	180	4.29
	Subtotal	*584*	*5.39*	*412*	*9.83*
Total		**10,842**	**100.0**	**4193**	**100.0**

Table 8
Incidence counts of paragraphs containing placekeeping at the right margin (TPs only).

	Types of paragraph	TPs		
		Count (A)	Total (B)	A/B (%)
Step	Action step	1341	1731	77.5
	Referencing (step)	12	12	100.0
	Conditional action step	100	231	43.3
	Continuous action step	6	6	100.0
Related with steps	Record row	13	159	8.2
	List element	816	1060	77.0
Others	Information	129	180	71.7

- All the paragraphs of the continuous action step type starting with **WHILE** (95 for OPs and 6 for TPs) did not contain **THEN**; and no paragraphs of the continuous action step type starting with **IF AT ANY TIME** were found.

In Table 9, incidence counts in parentheses are for those paragraphs that had any of the syntactic elements, **IF**, **WHEN**, **WHILE**, **IF AT ANY TIME**, or **THEN**, in non-emphasized form. Those counts in parentheses were summed up to each corresponding count prior to the parentheses. It is recommended to emphasize them in consistent style for better readability and easy recognition of such conditional work flows.

4.2.4. *Improperly written paragraphs of the information type in the instructions section*

According to the paragraph classification method introduced in section 3.3.2, there should be no paragraphs classified as the information type in the *instructions* section. However, a total of 270 paragraphs were classified as this type as shown in Table 7. Those were improperly written step statements or NCW statements [2].

The *iExtractor* classifies those paragraphs (or just *action* clauses of those paragraphs starting with any of the conditional or continuous components, such as **IF**, **WHEN**, **WHILE**, or **IF AT ANY TIME,** whether being emphasized or not) into one of the four types shown in Table 10. Incidence counts in parentheses, in Table 10, are for the *action* clauses that are summed up to each corresponding count prior to the parentheses. None of those four types had proper action verbs.

- For the first three types, it is recommended to revise each of those paragraphs into (i) an imperative style with the same context, (ii) a clear directive from the procedure user's perspective employing a proper action verb, like '**NOTIFY** < *nominative(s)*> to <*verb*> ~' or '**ENSURE** that ~' or, (iii) an NCW statement preceded with a proper NCW heading.
- For each paragraph of the last type, it is recommended to review its purpose carefully, and then to rewrite it properly according to its intended purpose.

4.3. *Step component identification results*

As explained in section 3.3.3, a step statement is decomposed into six components. The core components of step statements, the action verb and the target object, could be classified further by their appearance counts in each step statement. Table 11 shows those detailed types of core components and the number of step paragraphs corresponding to each type.

- As anticipated, step paragraphs of the first type, in Table 11, make up the majority, 90.9% for OPs and 84.75% for TPs.
- Steps having two-related action verbs are allowed, however, steps having more than one target objects are not recommended [2]: Each of paragraphs of types 2 and 5 needs to be reviewed to assure there exist no configuration dependency between two target objects; For paragraphs of types 3 and 6 (there is only one such paragraph), it is advised to list their target objects in bullet form below the step or to split them into multiple step statements.
- For paragraphs of types 7 or 8, it is also recommended to split them into multiple step statements.

Table 12 shows the top five frequently used two-related action verbs sharing the same target object(s), found from this analysis.

Table 13 and Table 14 show the incidence count (and the incidence ratio percentage in parenthesis) of each optional component for each step type, respectively for OPs and for TPs. Based on the incidence ratio percentages, both tables show similar results. It was also revealed that the supporting information component appeared frequently for all step types.

The procedure writing guide of this specific NPP site deviates slightly from some of the PPA's formatting recommendations. When PPA guidance was introduced, this site made the determination that aligning the site's writing guide with every formatting detail of the PPA would, in some cases, require extensive effort with only minor benefits. In addition, it was recognized that there were significant human factor advantages in maintaining a high priority for a consistent procedure format. Currently, this plant is migrating to an advanced procedure program that will integrate with digital technology. All the current procedures will require conversion into this new program, providing a structured opportunity to more closely align the new procedure program to PPA guidelines. This site will be able to use the results of this analysis to identify latent weaknesses, in structure and presentation, existing in procedures.

5. Concluding remarks

This paper introduces a newly developed methodology and the software, called *iExtractor*, which automatically captures the syntactic and semantic information from NPP procedures utilizing natural language processing-based technologies. The *iExtractor* adopts the rule-based information extraction approach integrated with a newly developed ontology for NPPs to enhance its effectiveness. Detailed steps of the *iExtractor* were described with

Table 9
Incidence count in detail for paragraphs of the types of conditional action step or continuous action step.

Step type	Detailed type	OPs	TPs
Conditional action step	**IF** < *condition(s)*>, **THEN** <*action(s)*>	819	175
	IF < *condition(s)*>, <*action(s)*>	3 (3)	14 (14)
	WHEN < *condition(s)*>, **THEN** <*action(s)*>	6	–
	WHEN < *condition(s)*>, <*action(s)*>	275 (7)	42
Continuous action step	**WHILE** < *condition(s)*>, **THEN** <*action(s)*>	–	–
	WHILE < *condition(s)*>, <*action(s)*>	95 (7)	6

Table 10
Detailed types of improperly written paragraphs of the information type.

Detailed type		OPs	TPs
Sentence (or clause)	In passive voice	40 (11)	36 (4)
	With nominative-associated verb phrase(s)	39 (10)	124 (8)
	Multiple sentences each as one of the above types	6 (1)	–
Simple phrase		5	20
Total		90 (22)	180 (12)

Table 11
Detailed types of core components and their incidences.

Detailed type		OPs		TPs	
		Count	%	Count	%
1	Single AV and single TO	4767	90.90	1678	84.75
2	Single AV and two TOs	136	2.59	84	4.24
3	Single AV and more than two TOs	–	–	1	0.05
4	Two-related AVs and single TO	90	1.72	72	3.64
5	Two-related AVs and two TOs	–	–	33	1.67
6	Two-related AVs and more than two TOs	–	–	–	–
7	Multiple occurrences of any of the above types at a sentence	210	4.00	83	4.19
8	Multiple sentences and at least one of them is any of the above types	41	0.78	29	1.46
Total		5244	100	1980	100

(AV: action verb, TO: target object).

Table 12
Top five frequently used two-related action verbs.

OPs		TPs		All	
Action verb	Count	Action verb	Count	Action verb	Count
Close and lock	15	Sign and record	33	Close and lock	37
Depress and hold	10	Close and lock	22	Sign and record	33
Open and lock	8	Close and time	13	Depress and hold	20
Place and hold	7	Depress and hold	10	Close and time	13
Establish or maintain	6	Measure and record	10	Open and lock	11

Table 13
Incidence of each optional component for each step type (OPs only).

Step type	Total count	Condition	Critical Information	Adverb	Supporting Information
Action step	4022	–	78 (1.94%)	150 (3.73%)	2768 (68.82%)
Branching (step)	2	–	–	–	–
Referencing (step)	22	–	1 (4.55%)	–	14 (63.64%)
Conditional action step	1103	1103 (100%)	–	36 (3.26%)	486 (44.06%)
Continuous action step	95	95 (100%)	–	4 (4.21%)	55 (57.89%)

Table 14
Incidence of each optional component for each step type (TPs only).

Step type	Total count	Condition	Critical Information	Adverb	Supporting Information
Action step	1731	–	15 (0.87%)	44 (2.54%)	1227 (70.88%)
Referencing (step)	12	–	–	–	6 (50%)
Conditional action step	231	231 (100%)	–	13 (5.63%)	80 (34.63%)
Continuous action step	6	6 (100%)	–	–	3 (50%)

examples in three phases, preprocessing, natural language processing, and the main information extraction phase. The *iExtractor* has unique features for extracting more valid information from NPP procedures, such as enhancement of POS tagging, classification of paragraph types, and identification of step statement components, each integrated with its own rule set. The *iExtractor* system has been fully implemented in Microsoft C# utilizing some publicly available components at the first two phases. It supports all steps of the proposed method and stores all the extracted information in various forms including database tables. Case study results of the *iExtractor* system obtained from twenty-five procedures of a U.S. commercial NPP, ten operating procedures and fifteen testing procedures, were also introduced. Selected findings associated with the relevant summary statistics provided by the *iExtractor* system were introduced.

This paper did not introduce a systematic evaluation of the information extraction results of the proposed method. Various metrics to evaluate information extraction results have been proposed [16,18,48]; however, they all require domain experts' manual annotation for the selected data set, before or after information extraction is applied. Without any gold standard in a specific domain, such evaluation results could be subjective [49]. Instead, during the case study, more effort was devoted to improving various outputs of the *iExtractor* system to detect any false or missed results in more convenient ways, *e.g.*, a list of semantic entities annotated by rules for each origin instance entity included in the ontology and a list of non-annotated (compound) nouns. On average, a one hundred-page procedure was processed in minutes. Preprocessing of the procedure document files takes time, but it is required only once. Based on our experience, the amount of false or missed information was limited after several iterations of reflecting feedbacks including those on plant-specific procedure writing styles. A more specific evaluation on the causal relationships among the stepwise performances of the NLP-based applications is proposed as a topic for further study.

The *iExtractor* system may make it easier to develop and maintain sound and effective procedures and help to reduce the workload involved in their management. It can provide overall comprehension of the analyzed procedures and indicate parts of procedures that need improvement. The rich information extracted from procedures could be utilized as the basis for their enhanced management. A higher level of oversight could be challenging that may include causal relationships among the performances at each stage of the NLP-based applications, verification and validation of procedures in terms of their compliance with detailed procedure writing guidelines, improvement of integrity with other configuration information components, and introduction of more advanced procedure management systems integrated with digital technologies [50].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This study was supported in part by Korea Hydro & Nuclear Power Co., Ltd., Republic of Korea. (No. 17-Tech-14).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.net.2020.08.010.

References

[1] International Atomic Energy Agency, Good Practices with Respect to the Development and Use of Nuclear Power Plant Procedures, IAEA-TECDOC-1058, Vienna, 1998.

[2] Procedure Professionals Association, Procedure Writer's Manual, PPA AP-907-005, 2016. Revision 2.

[3] U.S. Department of Energy, Writer's Guide for Technical Procedures, DOE-STD-1029-92, Revision 0, 1998.

[4] Procedure Professionals Association, Functional Requirements for Advanced and Adaptive Smart Documents, PPA AP-907-005.001, 2017. Revision 0.

[5] S.C. Peres, N. Quddus, P. Kannan, L. Ahmed, P. Ritchey, W. Johnson, S. Rahmani, M.S. Mannan, A summary and synthesis of procedural regulations and standards - informing a procedures writer's guide, J. Loss Prev. Process. Ind. 44 (2016) 726–734, https://doi.org/10.1016/j.jlp.2016.08.003.

[6] K. Thomas, B. Hallbert, Long-term Instrumentation, Information, and Vontrol Systems (II&C) Modernization Future Vision and Strategy, INL/EXT-11-24154, Revision 2, Idaho National Laboratory, 2013.

[7] P.H. Seong, H.G. Kang, M.G. Na, J.H. Kim, G. Heo, Y. Jung, Advanced MMIS toward substantial reduction in human errors in NPPs, Nucl. Eng. Technol. 45 (2013) 125–140, https://doi.org/10.5516/NET.04.2013.700.

[8] S.J. Lee, P.H. Seong, Design of an integrated operator support system for advanced NPP MCRs: issues and perspectives, in: H. Yoshikawa, Z. Zhang (Eds.), Prog. Nucl. Saf. Symbiosis Sustain. Adv. Digit. Instrumentation, Control Inf. Syst. Nucl. Power Plants, Springer Japan, Tokyo, 2014, pp. 11–26, https://doi.org/10.1007/978-4-431-54610-8_2.

[9] International Atomic Energy Agency, Information Technology for Nuclear Power Plant Configuration Management, 2010. IAEA-TECDOC-1651.

[10] Procedure Professionals Association, Procedure Process Description, 2016. PPA AP-907-001, Revision 2.

[11] J. Cowie, Y. Wilks, Information extraction, Commun. ACM 39 (1996) 80–91.

[12] S. Sarawagi, Information extraction, found, Trends Databases. 1 (2008) 261–377.

[13] M. Mannai, W.B.A. Karâa, H.H. Ben Ghezala, Information extraction approaches: a survey, in: D.K. Mishra, A.T. Azar, A. Joshi (Eds.), Inf. Commun. Technol., Springer Singapore, Singapore, 2018, pp. 289–297.

[14] D.E. Appelt, B. Onyshkevych, The common pattern specification language, in: Proc. A Work. Held Balt. Maryl. Oct. 13-15, 1998, Association for Computational Linguistics, USA, 1998, pp. 23–30, https://doi.org/10.3115/1119089.1119095.

[15] H. Cunningham, D. Maynard, V. Tablan, Jape: a Java Annotation Patterns Engine, Univ. of Sheffield, Department of Computer Science, 2000. Technical report CS–00–10.

[16] J. Zhang, N.M. El-Gohary, Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking, J. Comput. Civ. Eng. 30 (2016) 4015014.

[17] N. Ireson, F. Ciravegna, M.E. Califf, D. Freitag, N. Kushmerick, A. Lavelli, Evaluating machine learning for information extraction, Proc. 22nd Int. Conf. Mach. Learn. (2005) 345–352.

[18] P. Zhou, N. El-Gohary, Ontology-based automated information extraction from building energy conservation codes, Autom. ConStruct. 74 (2017) 103–117, https://doi.org/10.1016/j.autcon.2016.09.004.

[19] W. Wong, W. Liu, M. Bennamoun, Ontology learning from text: a look back and into the future, ACM Comput. Surv. 44 (2012).

[20] N.F. Noy, D.L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, KSL-01-05, Stanford knowledge systems laboratory, Stanford, CA, 2001.

[21] C. Arora, M. Sabetzadeh, L. Briand, F. Zimmer, Automated checking of conformance to requirements templates using natural language processing, IEEE Trans. Software Eng. 41 (2015) 944–968.

[22] Y. Zhao, X. Diao, J. Huang, C. Smidts, Automated identification of causal relationships in nuclear power plant event reports, Nucl. Technol. 205 (2019) 1021–1034, https://doi.org/10.1080/00295450.2019.1580967.

[23] A. Jain, M. Ganesamoorty, NukeBERT: a Pre-trained Language Model for Low Resource Nuclear Domain, 2020. ArXiv Prepr. ArXiv2003.13821.

[24] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018. ArXiv Prepr. ArXiv1810.04805.

[25] P. Shi, J. Huo, Q. Wang, Constructing ontology for knowledge sharing of materials failure analysis, Data Sci. J. 12 (2014) 181–190.

[26] N.M. Meenachi, M.S. Baba, Development of semantic web-based knowledge management for nuclear reactor (KMNuR) portal, DESIDOC J. Libr. Inf. Technol. 34 (2014) 426–434.

[27] A. Pruttianan, N. Lau, V. Tech, S. Anders, M.B. Weinger, Ontology to guide scenario design to evaluate new technologies for control room modernization, in: Nucl. Plant Instrument, Control Hum. Mach. Interface Technol., 2017, pp. 206–214.

[28] R. Elhdad, N. Chilamkurti, T. Torabi, An ontology-based framework for process monitoring and maintenance in petroleum plant, J. Loss Prev. Process. Ind. 26 (2013) 104–116.

[29] T. Sobral, T. Galvão, J. Borges, An ontology-based approach to knowledge-assisted integration and visualization of urban mobility data, Expert Syst. Appl. 150 (2020) 113260, https://doi.org/10.1016/j.eswa.2020.113260.

[30] Y. Wu, V. Ebrahimipour, S. Yacout, Ontology-based modeling of aircraft to support maintenance management system, IIE Annu. Conf. Proc. (2014) 1159–1168.

[31] R.C. Ward, A. Sorokine, Nuclear Power Plant Ontology in OWL Format, ORNL/LTR-2012/467, Oak Ridge National Laboratory (ORNL), U.S. Department of Energy, 2012.

[32] M.A. Musen, The protégé project: a look back and a look forward, AI Matters 1 (2015) 4–12.

[33] Gate, ANNIE: a Nearly-New Information Extraction System, 2018. https://gate.ac.uk/ie/annie.htm.

[34] Stanford NLP Group, Stanford Named Entity Recognizer NER. https://nlp.stanford.edu/software/CRF-NER.shtml, 2018.

[35] Microsoft, NET API, 2018. https://docs.microsoft.com/en-us/dotnet/api/microsoft.office.interop.word?view=word-pia.

[36] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Syst. Demonstr., 2014, pp. 55–60.

[37] H. Ryu, Open Korean Text Processor, 2018. https://github.com/open-korean-text/open-korean-text.

[38] C.D. Manning, C.D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, MIT press, 1999.

[39] D. Jurafsky, J.H. Martin, Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, second ed., Prentice-Hall, Upper Saddle River, NJ, 2009.

[40] M.P. Marcus, M.A. Marcinkiewicz, B. Santorini, Building a large annotated corpus of English: the penn treebank, comput, Linguist 19 (1993) 313–330.

[41] M.-C. De Marneffe, C.D. Manning, Stanford Typed Dependencies Manual, Technical Report, Stanford University, 2008.

[42] D. Sarkar, Text Analytics with Python: a Practical Real-World Approach to Gaining Actionable Insights from Your Data, Apress, 2016.

[43] P. Paroubek, Evaluating part-of-speech tagging and parsing, in: Eval. Text Speech Syst., Springer, 2007, pp. 99–124.

[44] G.A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (1995) 39–41.

[45] U.S. Nuclear Regulatory Commission, APR1400 Design Control Document and Environmental Report, 2018. https://www.nrc.gov/reactors/new-reactors/design-cert/apr1400/dcd.html.

[46] Korea Hydro, Nuclear Power, Glossary of Nuclear Power, 2017.

[47] J. Oxstrand, K. LeBlanc, Computer-based Procedure for Field Activities: Results from Three Evaluations at Nuclear Power Plants, Idaho National Laboratory, Idaho Falls, ID (United States), 2014.

[48] J. Piskorski, R. Yangarber, Information extraction: past, present and future, in: Multi-Source, Multiling. Inf. Extr. Summ., Springer, 2013, pp. 23–49.

[49] D.C. Wimalasuriya, D. Dou, Ontology-based information extraction: an introduction and a survey of current approaches, J. Inf. Sci. 36 (2010) 306–323, https://doi.org/10.1177/0165551509360123.

[50] Electric Power Research Institute, Improving the Execution and Productivity of Maintenance with Electronic Work Packages: a Mobile Work Management Initiative, 2015.