Towards Extraction of Validation Rules from OPC UA Companion Specifications

Nilay Tufek[1], Aparna Sai Sree Thuluva[2], Valentin Philipp Just[3], and Marta Sabou[4]

[1]Technology Department, Siemens AG, Munich, Germany; nilay.tuefek-oezkaya@siemens.com
[2]Technology Department, Siemens AG, Munich, Germany; aparna.thuluva@siemens.com
[3]Inst. Computer Engineering, Vienna University of Technology; valentin.just@tuwien.ac.at
[4]Inst. Data, Process Knowledge Management, Vienna University of Economics and Business; marta.sabou@wu.ac.at

Abstract—**Interoperability is the key to industrial automation. OPC UA aims to provide interoperability between industrial machines at the network, protocol, and semantic layers. In order to achieve it, the implementation of a machine and its nodeset file should comply with the OPC UA standard. However, there are a few areas for improvement in technology to automatically check compliance since the compliance rules/validation rules are expressed in textual format in specification documents. In other words, they are not machine-interpretable. Converting the text-based specification into machine-readable form is a complex problem since each companion specification is domain-specific and written by a set of industry experts from all over the world from diverse language backgrounds. The specifications are majorly unique, with little commonality between them. Because of these reasons, it is challenging to develop a generic information extraction approach to retrieve rules that work for all specifications. In this study, we aim to handle this challenge to extract the textual rules from specifications automatically by applying Natural Language Processing and Machine Learning technologies. We present our methodology based on Named Entity Recognition for OPC UA documents for information extraction and text classification to identify rules. To achieve this goal, we created five named entity data sets from five selected OPC UA companion specifications. The sentences in the data set and the entities in the sentences were annotated by two OPC UA experts. We point out a deeper analysis of the data sets to highlight common and unique entities in them and show their usage in identifying the rules in the specifications.**

Index Terms—**OPC UA, Named Entity Recognition, Natural Language Processing, Classification**

I. Introduction

A factory shop floor has various machines with different capabilities and skills from multiple vendors. The objective is to have these complex machines interact with one another to achieve a common goal, such as automating a production task. However, if these machines utilize different protocols and present their data and meanings using distinct information models, enabling interaction and integration between machines would be a tedious and time-consuming process [1]. Therefore, implementing custom solutions under such circumstances would not be efficient. Standards such as OPC UA are considered to unify these various machines and provide a common interface or language for them to interoperate with each other [2]. However, if the machine implementation is not compliant with the standard, we repeat the same problem, as mentioned before. Thus, validating a machine's implementation against the standard for interoperability is essential for achieving automation. The OPC UA standard is complex, and its information model and semantics are rich and challenging to understand, model, implement and validate manually. There are two issues that arise from this situation. Firstly, the *nodeset* files of the machines become non-compliant. A nodeset file is an instantiation of the chosen OPC UA information model(s) for the machine. Secondly, the server implementation becomes non-compliant, further hindering the machines' interoperability on the shop floor.

Existing compliance-checking tools can validate the implementation of an OPC server against the core specification. The companion specification validation is provided only for a few specifications[1]. The existing tools to validate a nodeset file (to some extent) based only on the core specification[2]. To the best of our knowledge, there are no tools to validate a nodeset file based on the semantics defined in companion specifications. Therefore, the validation process for some companion specifications is performed manually. This is neither a feasible nor a scalable process as the number of companion specifications is rapidly increasing in tandem with the use of OPC UA in the industry. Moreover, the availability of OPC UA experts who can understand and implement the validation for each companion specification is limited. To conclude: there is currently a gap in technology that can automate the validation of industrial machines against the OPC UA companion specifications.

We aim to fill this gap by (semi-) automating such validation processes. In previous work [3], we presented an overarching validation process based on emerging Natural Language Processing (NLP), Machine Learning (ML), and Semantic Web (SW) technologies. The idea of that work was to extract the rules expressed in textual companion specifications and formalise them as validation rules, which can be used to automate the information model of individual machines and

[1] https://opcfoundation.org/news/press-releases/opc-ua-compliance-test-tool-uactt-extended-for-companion-spec-validation/

[2] https://support.industry.siemens.com/cs/document/109755133/siemens-opc-ua-modeling-editor-(siome)?lc=en-de

to verify the runtime behavior of the server. A key challenge was providing effective methods to predict which sentences in the textual documentation contained information that should be formalised as rules. In this paper, we focus on this particular task. To that end, as a basis for deciding which sentence potentially describes a rule, we require a better understanding of the text itself in terms of key entities that occur in those texts. Such key entities are referred to, in the area of NLP, as named entities (NEs) and the process of identifying them is known as Named Entity Recognition (NER). In particular, both gold standard data sets of NEs and corresponding NER algorithms are needed. As these artifacts are currently missing for OPC UA, our **research questions** are:

- RQ1: How to create a gold standard NE data set for OPC UA? How to perform the annotation process in a methodologically correct way?
- RQ2: What are the baseline algorithms for NER in the domain of OPC UA which can be created based on the gold standard data set derived at RQ1? What are the performance levels that can be achieved in this specialized domain?
- RQ3: To what extent can NEs be used to support the process of classifying sentences in terms of whether they represent rules? Which classification algorithm can be used and what is its performance?

The main **contribution** of this article is threefold, as defined below:

(i) A manually annotated, *gold standard NE data set* for five OPC UA companion specifications. The creation of this data set followed the best practices of corpus annotation in NLP [4].
(ii) *OPC UA specific algorithms for NER and sentence classification* developed based on the insights gained during the annotation process of the NE data set.
(iii) *Performance evaluation* of the proposed algorithms.

The paper is organized into several sections. Section II provides an in-depth overview of OPC UA and its companion specifications and a discussion of the validation problem. Section III outlines the NLP methodology we developed to extract rules from text, including a presentation of the guidelines for developing and annotating data sets. Section IV, which is the result section, highlights the differences and statistics in the data sets created based on the five OPC UA companion specifications, and we share our findings and outcomes of the implemented algorithms. Finally, in Section VI, we summarise the current work and discuss potential future research directions.

II. Background and Problem Definition

A. Background and Motivating Use cases

In order to elaborate on the challenges and obstacles related to implementing an OPC UA companion specification and ensure its adherence, a motivating use case is presented from the perspectives of various stakeholders engaged in this process. These stakeholders participate in both the development of companion specifications and the application of these specifications in designing and executing machinery. The stakeholders are as follows:

a) The OPC Foundation/ OPC community which develops the companion specifications: A companion specification consists of a textual document in PDF format and an information model in XML and CSV format. Experts develop it from different parts of the world with different language backgrounds (many of them may not be native English speakers). Moreover, each specification focuses on a specific domain (e.g., food and beverages, packaging technology, robotics etc.). Because of the significant variations between the specifications, there is a high degree of heterogeneity in the way they are written and structured. Applying NLP and AI algorithms to them for efficient information extraction is challenging. For practical information extraction, the specification documents need a better structure and templates for expressing rules in a unified way in the text. This work also aims to contribute best practices and guidelines to the OPC Foundation to structure the companion specifications.

b) A machine builder/ system integrator who uses the companion specifications: As briefly presented in Section I a machine builder should use a companion specification document and its information model to implement the machine to be interoperable. For this purpose, they develop the node-set file by instantiating the information model provided by the specification and implement the server according to the specification. However, in this process, they encounter several problems. OPC UA has a deeper knowledge requirement, it is time-consuming, and there is no efficient tooling available while only a few experts are available. The development of effective validation tools is essential in addressing these issues. Thus, our goal in this study is to create tools that can assist machine builders/system integrators in the validation process.

B. Problem Definition

The problem can be handled in two layers. The first one is the general reasons for our motivation, which are: (i) Increasing the use of OPC UA requires validation steps to exist or be automated. (ii) The time and number of experts in the domain are limited. (iii) That is an error-prone process for human-being, including spelling mistakes, wording changes, etc. These identified problems are based on our own experience and observations of the industry's needs described in [3]. Therefore, an end-to-end solution is designed to extract information from natural language-based PDF texts and generate SPARQL queries. An example of this end-to-end process can be found in Table 1. Each sub-process in these thoroughly evaluated stages is a research topic in its own right. In this project's approach, many technical problems were addressed and resolved. These technical-level problems and challenges can be summarised as follows: (i) The need for a new labelled data set. (ii) The text we are working on belongs to a particular field and has unique terms not used elsewhere, (iii) Each companion specification which has

TABLE I
AN EXAMPLE OF INFORMATION EXTRACTION FROM TEXT.

Text from PDF	Constraint Sentence	Constraint Type	Named Entities	Rule in SFN	Rule in SPARQL

All DataType that are structures include ”DataType” as part of the name, this is to be able to differentiate from any VariableTypes that will just end in Type.	yes	Information Model Constraint	All,
			DataTypes,
			structures,
			include,
			”DataType”,
			VariableTypes
				A node is of type DataType. The node has a BrowseName attribute. The BrowseName of the node should include the string “DataType” in it.	SELECT ?node WHERE
					{
					?node opcua:nodeClass ”64”ˆˆxsd:int.
					?node opcua:nodeId ?nodeId .
					FILTER (REGEX(STR(?nodeId),
					”ˆhttp://opcfoundation.org/UA/PackML/”))
					?node opcua:browseName ?browseName .
					FILTER (REGEX(STR(?browseName),
					”ˆ((?!DataType).)∗”))
					}

its own vocabulary that should be discovered. Therefore, by considering these obstacles, creating a new entity terminology specifically for OPC UA documents is inevitable. Furthermore, novel NER and classification algorithms should be developed to proceed with this complex data effectively.

III. METHODOLOGY

Our objective is to propose a solution for efficiently extracting information regarding validation rule generation from OPC UA companion specifications. In line with this purpose, as the baseline, we worked on Robotics, PackML, MachineTools, PROFINET, and Weihenstephan specifications. These documents contain a significant amount of domain-specific information, which is not adequately captured by state-of-the-art pre-trained information extraction algorithms commonly used for generic NLP-based textual data. Our preliminary experiments have shown that those pre-trained algorithms [5] yield similar results to random entity selection algorithms. Therefore, we propose the development of novel and domain-specific algorithms to effectively extract the unique and document-specific entities that exist within each document. The ultimate goal of this study is to create a generic solution that can be applied to other companion specifications, providing engineers with an efficient means of identifying the rules within the document.

The end-to-end proposal from PDF document to sentence classification is shown in Figure 1. Regarding this design, the primary input is a PDF document. In the manual preprocessing step the sentences from the PDF are extracted and placed in an MS Excel file. The invalid sentences such as incomplete sentences are removed at the *preprocessing* step (this step would be automated in the future). After that, NEs are extracted using related information models in XML format. Then, using the extracted entities, classification algorithms are performed. In order to achieve this automated end-to-end process, there is another methodological approach behind it. This approach includes releasing the data sets, NER and Binary Classification (BC) processes efficiently. This iterative approach is demonstrated in Figure 1.

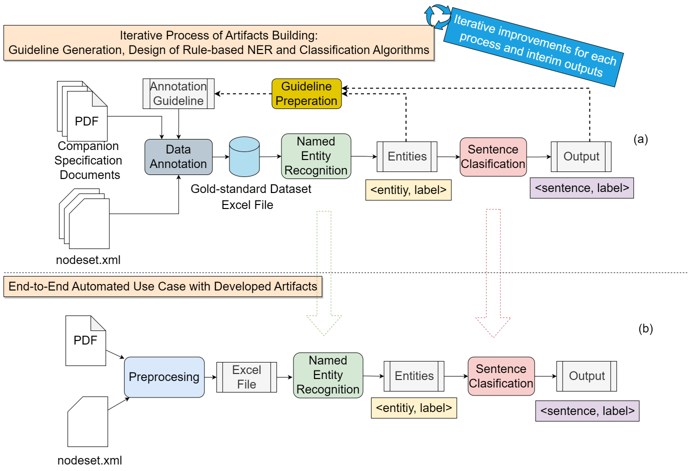


Fig. 1. Overall design decisions including NER and Binary Classification. (a) is an iterative process of the gold standard guideline generation, data set creation, and rule-based system designs. (b) is a usage of the data sets and the algorithms.

A. Data Set

The gold standard data set was prepared based on the guideline that we designed. We specified the amount of data that should be annotated, the entity categories, the rule-based NER algorithm and classification algorithms iteratively. We established the annotation principles before creating the gold standard data set. The guidelines and rules outlined in the book [4] were followed to determine the boundaries and limitations of the annotation process. This involved defining the project's purpose, specifying the number of companion documents to be worked on, determining the total number of required sentences, categorizing the entities, identifying whether an entity belonging to multiple categories or not, and determining whether an entity consisting of more than one word. In addition, we established sentence classification strategies and determined the number of annotators. These steps were carried out using the MAMA (Model-Annotate-Model-Annotate) cycle [4], which involves iterative processes of modeling/guiding, annotating, evaluating, and revising. The guide for creating the data set was developed based on the decisions made regarding the following topics:

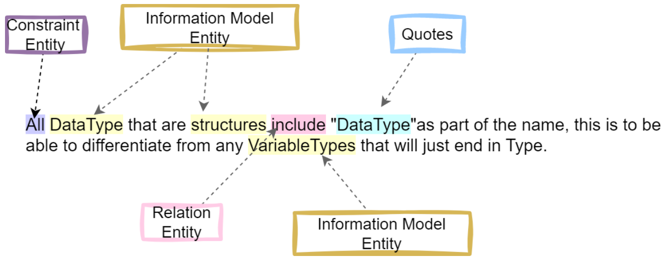
- **The goal:** The goal is to use the sentences to determine the entities and classify the sentences by constraints.
- **The data:** The gold standard data set is created us-

ing five companion specifications: PackML, Robotics, PROFINET, MachineTools and Weihenstephan.

- **The labels:** We extract the sentences from PDF file(s), and each sentence is annotated according to involving entities, entity categories and sentence classes. Those entities are Information Model Keywords, Constraint Keywords, Relation Keywords, Numbers, Quotes, and Runtime-only keywords. The entity categories are explained in detail in the next section. Additionally, there are two different sentence classification approaches. The first is classifying the sentence according to whether or not it contains a constraint, and the second is categorizing whether it is a runtime or information model constraint.
- **The Annotators:** The number of experts who annotate the sentences is two. We use Cohen's Kappa [6] metric to measure the degree of Inter-Annotator Agreement (IAA).
- **Tooling:** The annotation is performed using the Microsoft Excel platform.

B. Algorithms for Named Entity Recognition

In this scope, the most remarkable and effective words and phrases in the sentences were determined during the investigation and designing processes. It was observed that important entities could be classified into five categories. In order to find and label the entities, we developed a rule-based string comparison algorithm. Figure 2 depicts a sentence from the OPC UA specification with six NEs belonging to four of the five entity categories.



All DataType that are structures include "DataType"as part of the name, this is to be able to differentiate from any VariableTypes that will just end in Type.

Fig. 2. Example OPC UA specification sentence including named entities of different entity categories.

a) Information Model Entity: Each companion specification is released with an information model in XML format which formalizes the nodes defined in the companion specification. These nodes are considered as Information Model (IM) entities in our work. If such IM entities are found in a sentence then they are labeled as IM entities. While there can be an exact match between the entities, there are also outliers that we must catch. For example, an IM entity could a combination of multiple words in CamelCase form. However, the same entity may not be written in the sentence in camelCase format or it may be referred in plural form in the sentence and in the singular form in the information model.

b) Relation Entity: represents the relationship between two IM entities. In an information model, the relationship between nodes is modeled using well-defined references. However, in the specification documents, diverse words are used to represent their relationship. In this category, we aim to identify such keywords in a sentence. These entities play a key role in identifying the relation between IM entities and thus the constraint on them. A list of relation keywords is pre-defined using the existing English language corpus for this purpose and also based on our observations from the five chosen companion specifications.

c) Constraint Entity: represent the words in a sentence that describe a constraint on the IM and relation entities. These are words and phrases such as ”all”, ”at least”, ”must”, etc. Constraint keywords are identified in the same fashion as relation entities. In our algorithm, we match the single-word constraint entities in a sentence using the predefined list.

d) Quotes: The information in the quotation mark is always notable from a rule sentence perspective. Therefore, the strings represented in various quotation marks are extracted. For example: in rules where string matching is the constraint.

e) Numbers: The algorithm detects the numbers in the sentence on purpose by eliminating the irrelevant numbers such as the Figure and the Table numbers. On the other hand, numbers also define constraints such as allowed value range for a variable etc.

f) Runtime-only Entity: a rule can be a rule on the information model or a rule that should be validated during runtime on the server. In this category, we identify the words in a sentence which describe a runtime constraint. At this point, this entity is recognized but it is not used to classify a sentence into a runtime rule. This is subject to future work.

C. Algorithms for Sentence Classification

We designed a cascade decision tree for classification of a sentence. Thus, the first phase is to determine if a sentence includes a constraint; in other words, we find out if a sentence is a rule-sentence or a non-rule-sentence. In the scope of this paper, we focused on the BC phase of the classification vision. (Further classification steps will be done in the future.) Therefore, two different algorithms were designed as the baseline using the extracted entities. These algorithms are a Rule-based algorithm and an ML-based algorithm. Both classification methods are compatible with the general flow seen in Figure 1 and use the extracted entities differently.

a) Rule-based design: There are several classification methods for textual-based documents. However, our data set is unique and has customized technical information. Therefore, we designed a *white box* for classification which is called Rule-based Classification. The prepared data set from five companion specifications were used to build the algorithm shown in Algorithm 1.

b) ML-based design: In order to show the usage of the flow with a ML-based classifier approach, we employed one of ML-based classifiers: Support Vector Machine (SVM) with Radial Based Function (RBF). We exploit scikit-learn python libraries [7] for that. First of all, there is a feature extraction step needed. The input of that step is the extracted entities in five categories for each sentence and the output is a feature

Algorithm 1 Rule-based Classification

```
procedure RULEBASEDCLASSIFICATION($entities$)
    $pred\_labels \leftarrow []$
    for i = 0; i < len($entities$) do
        if $entities[i].information\_model$ is Empty then
            $pred\_labels.[i] \leftarrow 0$
        else if $entities[i]$ is NOT Empty then
            $pred\_labels.[i] \leftarrow 1$
        else
            $pred\_labels.[i] \leftarrow 0$
        end if
    end for
    return $pred\_labels$
end procedure
```

vector based on the number of the recognized entities for each category, as demonstrated in Figure 3.

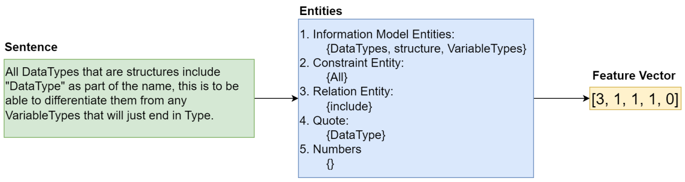

Sentence
All DataTypes that are structures include "DataType" as part of the name, this is to be able to differentiate them from any VariableTypes that will just end in Type.
Entities
1. Information Model Entities:
{DataTypes, structure, VariableTypes}
2. Constraint Entity:
{All}
3. Relation Entity:
{include}
4. Quote:
{DataType}
5. Numbers
{}
Feature Vector
[3, 1, 1, 1, 0]

Fig. 3. Handcrafted feature extraction using entities.

Using the feature vectors and the labels, the data is spitted into train and test. While training data consists of four specification documents such as PackML, Profinet, Weinstephan, MachineTools, the test data is Robotics. After the training is finished, the test data is used to get prediction results. The configuration of the models is used directly as used in [3].

IV. Results

A. Gold Standard Data Set

The gold standard data set was prepared and published in Hugging Face[3]. The data set consists of five sub-data sets, representing each companion specification document in MS Excel format. We analyze the data set from various aspects as follows:

1) Number of the sentences: We can begin with the quantitative analysis of the data set. Initially, we selected sentences from a document based on the applicable model and rule definition sections. This approach resulted in an unbalanced data set since each annotator labeled a different number of sentences. Annotator 1 annotated 1502 sentences, while Annotator 2 labeled only 446. Notably, the sentences that Annotator 2 annotated were also labeled by Annotator 1.

2) Occurrence of an entity in multiple documents: One of the most important indicators regarding the complexity of the data set is the distribution of the common entities in different companion specifications. As demonstrated in Figure 4, there are many unique entities compared to the common entities seen in multiple specification documents. That makes the generalization of NER for other specification documents even more challenging. It was analyzed and illustrated in the pie chart (Figure 4) for three categories: information model entities and relation and constraint entities.

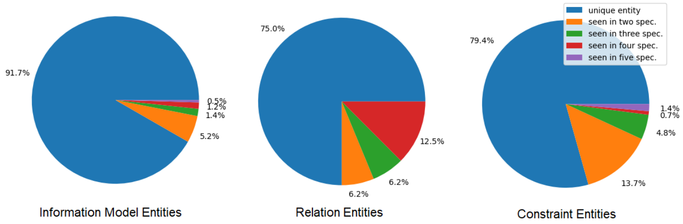


Fig. 4. The average of the distributions of the common entities among the documents by two annotators for three categories is shown.

3) The distribution of entities for each companion specification annotated by two annotators: Figure 5 illustrates the distribution of the entities in different entity categories for all companion specifications and each annotator. We can make three inferences from this graph: (i) Each companion specification displays an imbalanced distribution of entities across categories. (ii) Across all annotators and companion specifications, the most frequently occurring entities are related to information model keywords, with relational entity categories ranking second in frequency. (iii) Despite the varying number of annotated sentences, the proportion of labeled entities across categories is similar for both annotators.

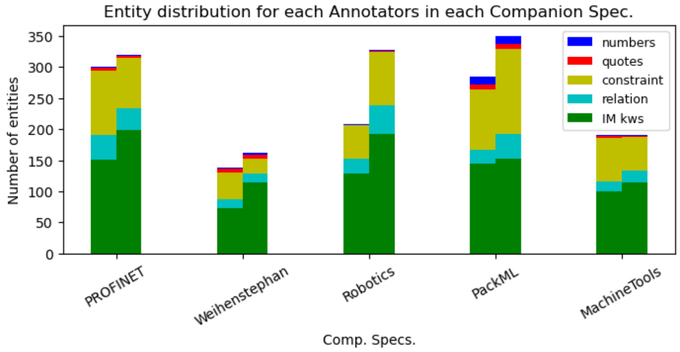


Fig. 5. The distribution of the entities in multiple specifications based on their categories, including *Information Model*, *Relation* and *Constraint*.

4) Analysis of entity occurrence: Besides the sentence amounts and distribution of the entity categories, we analyze the occurrence ratio of the entity itself in a companion specification. One example for PROFINET by Annotator 1 is seen in Figure 6. It is not only long-tail distribution, but also the variance is high, and this distribution trend is more or less the same for all companion specifications by both annotators.

Because of not dealing with the long tail, the first fifteen most common entities were considered for each specification in further demonstrations.

[3] https://huggingface.co/datasets/nilaytufek/OPC_UA_NER_BC

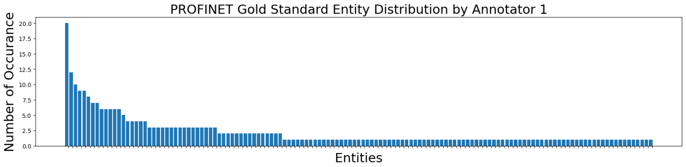


Fig. 6. Visualization of the occurrence ratio of each entity in PROFINET by Annotator 1.

5) The distribution of the most common fifteen entities for five companion specifications by each annotator: This part considers the *common annotated sentences* by two annotators. Figure 7 shows the most common entities for each specification by Annotator 1 and Annotator 2. It can be easily said that the distribution of the most frequent entities is different for each companion specification while they are some common entities. Therefore, it makes the data set even more complex. On the other hand, the distribution of entities for different annotators looks quite similar, which proves the consistency of the annotated data by different experts, as seen in Figure 7.

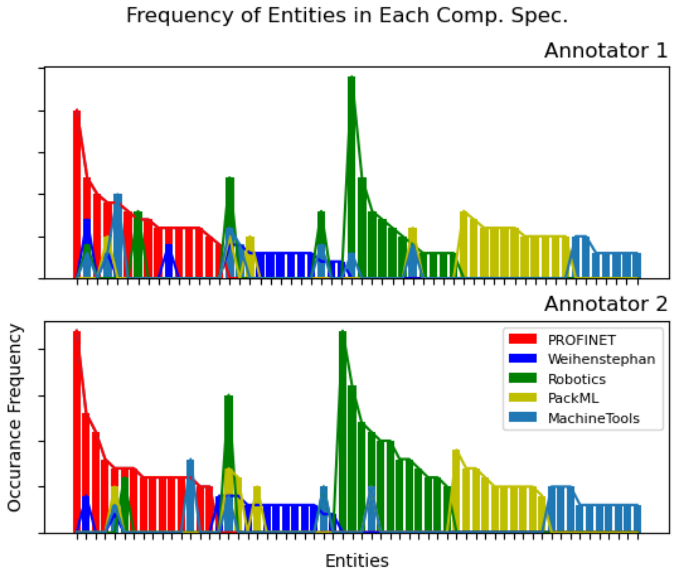


Fig. 7. The histogram of the most-common fifteen entities for each companion specification according to Annotator 1 and Annotator 2.

6) Intersection of NEs for five companion specifications: In order to calculate the correlation between annotators for entity labeling, we decided to use the *Intersection over Union (IoU)*. IoU is originally used to compute the intersection metric for image detection in the computer vision area by dividing the intersection area by the union area of the prediction and the ground truth. But it can also be used to determine the correlation and intersection area between two annotators in complex open-ended annotations, such as our entity labeling case in textual data. The IoU results are demonstrated in Table II.

$$IoU[i] = \frac{\text{the number of common entities in a sentence[i]}}{\text{the number of the union of entities in a sentence}}$$

Table II is generated using the average IoU of all samples of annotators for each specification document. The average IoU (AvrIoU) is calculated as follows:

$$AvrIoU = \frac{\sum_{i=1}^{\text{all}} \text{amount of common entities for sentence[i]}}{\sum_{i=1}^{\text{all}} \text{amount of union of entities for sentence[i]}}$$

Based on the common sense of the authorities, more than 0.5 is a good result [8]. All *AvrIoU* is calculated for each entity category and total. Based on the results, the correlations between two annotators for Information Model entities and Numbers and Quotes are fair. Considering the problem's complexity and the data itself, the other categories are also promising.

TABLE II
IoU Results of Two Annotators

	Information Model Entities	Relational Entities	Constraint Entities	Quotes	Numbers	Total
PROFINET	0.54	0.34	0.21	0.71	1.00	0.39
Weihenstephan	0.40	0.32	0.22	0.71	0.67	0.34
Robotics	0.54	0.26	0.18	1.00	1.00	0.38
PackML	0.57	0.32	0.27	0.78	1.00	0.41
MachineTools	0.63	0.30	0.21	0.67	1.00	0.43

7) Correlation between two annotators regarding sentence classification: In the previous section, the correlation of the entities is discussed. There is additional information regarding the sentence's BC. In order to measure the IAA, Cohen's Kappa Coefficient metric [6] is used. The results are shown in Table III.

TABLE III
Degree of Annotators' Agreement on five Comp. Specs.

	PROFINET	Weihenstephan	Robotics	PackML	MachineTools
Number of common sentences	77	84	94	103	82
Cohen's Kappa	0.46	0.49	0.49	0.39	0.35

Cohen suggested the Kappa result be interpreted as follows: values ≤ 0 as indicating no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement [9]. The IAA of our data set comes across *Moderate* interval.

B. Algorithm Results

1) Named Entity Recognition: The implementation of the NER algorithm was based on lexicons obtained from analyzing four companion specifications: *PackML, Weihenstephan, PROFINET, and MachineTools* (for the train set). The *Robotics* dataset was used to test the rule-based algorithm and evaluate its generalization effect. The performance of the algorithm was measured using the IoU metric between the ground truth and the algorithm's results. Table IV displays the results of both the train and test sets, as evaluated by Annotator 1 and Annotator 2. Additionally, a row for Random Entities was included to demonstrate the average IoU of randomly selected entities with Annotator 1 and Annotator 2. Therefore, we can interpret the table to understand the performance of the algorithm on both the train and test sets as well as its performance compared to random entities.

(i) In terms of all metrics, the baseline NER algorithm outperformed the results generated by the random entity generator applied to the sentence.

(ii) Even though the NER algorithm displayed better performance on the four companion specifications compared to the Robotics test data, there were no major discrepancies. This suggests that the algorithm has the potential to be applied to other companion specifications in a generalizable manner.

(iii) The similarity of the results from different annotators indicates that their trustworthiness is reasonable.

TABLE IV
NER - IoU Results

	PackML	Weihenstephan	PROFINET	MachineTools	Robotics
IoU for Annotator 1	0.24	0.23	0.26	0.25	0.21
IoU for Annotator 2	0.25	0.26	0.27	0.23	0.22
IoU for Random Entities	0.11	0.14	0.09	0.11	0.12

2) Binary Classification: Once entities have been extracted from sentences, one potential use case involves their utilization in classification. This study examines two primary approaches: a rule-based classifier (see Table V) and a ML-based classifier (see Table VI). Due to the presence of unbalanced data, the metric calculation strategy employed was the *Macro Average*.

TABLE V
Rule-based Classifier Results

Robotics\Macro Avrg	Precision	Recall	Accuracy	F1-Score
Annotator 1	0.49	0.49	0.46	0.44
Annotator 2	0.46	0.47	0.42	0.43
Random NER-based	0.19	0.22	0.23	0.22

Based on the results in the Tables V andVI, it can be said that:

(i) The ML-based Classifier outperformed the Rule-based Classifier, likely due to the complexity and dimensionality of the inputs (i.e., entities) involved.

(ii) In both algorithms, the results generated by Annotator 1 and Annotator 2 are similar, but Annotator 1's results are slightly better.

(iii) The last column of the tables is about the ML-based classification result based on the random entities. It is obvious that well-extracted entities help to improve the classification results.

C. Discussion

We conducted experiments on a limited dataset that OPC UA experts annotated. Despite the limited data and complexity, our approach using four specifications yielded positive results when applied to the fifth specification (Robotics). This shows that our rule extraction solution can be extended to other specifications with reasonable accuracy.

TABLE VI
ML-based Classifier Results

Robotics\Macro Avrg	Precision	Recall	Accuracy	F1-Score
Annotator 1	0.57	0.57	0.57	0.57
Annotator 2	0.55	0.55	0.52	0.53
Random NER-based	0.21	0.23	0.23	0.20

V. Related Work

We investigated several studies in the literature regarding our field of research. However, there is yet to be an exact end-to-end solution we need. Therefore, we investigate subtopics and similar research areas to ours.

Information Extraction from Textual Documents: Information extraction (IE) takes natural language-based text as input and produces detailed [10]. The difficulty of the process is relevant to the complexity of the data [11]. The closer the data is to natural language, the greater the required effort to analyze.

Named Entity Recognition: NER systems have been developed using hand-made rules, ML techniques [12], and deep learning networks [13]. In the state-of-the-art, deep learning methods such as transformers give the best results on very-large data sets [13] for generic entities. However, our data set is unique. Therefore, more than the pre-trained methods and state-of-the-art approaches is required. Nevertheless, transformers fail when the training data is limited, such as ours.

Sentence Classification: Many feature extraction and sentence classification methods have been invented so far. In text classifications, bag-of-words is a popular fixed-length feature [14] such as TF-IDF. However, more is needed for big corpora. Therefore, advanced methods such as word2vec [15] and word embedding techniques are developed.

Data sets: In this study's scope, we are releasing a new NER and sentence classification data set. Many data sets are released for NER purposes, such as WikiNER [16] data set, which is prepared for identifying and classifying mentions of people, organizations, locations and other NEs within the text. There are many data sets is a release for this purpose for different languages, such as IndoNLU [17], WNUT [18], and SBIC [19]. While most NLP data sets are focused on the extensive data set and generic labels, there are also a few domain-specific named entity data sets such as COVID-19 Open Research data set Challenge (CORD-19) [20]. Based on our knowledge, no NER data set has been prepared for industrial standards yet.

Finally, there needs to be a working solution for our problem and a prepared data set in the literature.

VI. Conclusion and Future Work

In parallel with OPC UA, the number and usage of companion specifications are increasing. Therefore, developing and validating information models compatible with specifications has become increasingly important. Within this project's scope, we conducted a given study on extracting compliance rules from textual documents. Accordingly, we prepared a gold standard dataset and NER and classification studies on this dataset. We showed the effective outputs of the analyses and the algorithms.

In the following steps, we plan to improve the NER and classification algorithms for all companion specifications using state-of-the-art methods, including deep learning. Moreover, we want to investigate tools like ChatGPT [21] to see how to exploit them. Finally, from now on, we expect that the other researchers and we will continue to work on these data sets and related results.

References

[1] Marcela Hernandez-de-Menendez et al. "Competencies for industry 4.0". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 14 (2020), pp. 1511–1524.

[2] Michael H Schwarz and Josef Börcsök. "A survey on OPC and OPC-UA: About the standard, developments and investigations". In: *2013 XXIV International Conference on Information, Communication and Automation Technologies (ICAT)*. IEEE. 2013, pp. 1–6.

[3] Yashoda Saisree Bareedu et al. "Deriving Semantic Validation Rules from Industrial Standards: an OPC UA Study". In: (2022).

[4] James Pustejovsky and Amber Stubbs. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications.* " O'Reilly Media, Inc.", 2012.

[5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[6] Matthijs J Warrens. "Five ways to look at Cohen's kappa". In: *Journal of Psychology & Psychotherapy* 5.4 (2015), p. 1.

[7] Lars Buitinck et al. "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[8] Alireza Fathi et al. "Semantic instance segmentation via deep metric learning". In: *arXiv preprint arXiv:1703.10277* (2017).

[9] Mary L McHugh. "Interrater reliability: the kappa statistic". In: *Biochemia medica* 22.3 (2012), pp. 276–282.

[10] Jing Jiang. "Information extraction from text". In: *Mining text data* (2012), pp. 11–41.

[11] Ela Kumar. *Natural language processing*. IK International Pvt Ltd, 2013.

[12] David Nadeau and Satoshi Sekine. "A survey of named entity recognition and classification". In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

[13] Jing Li et al. "A survey on deep learning for named entity recognition". In: *IEEE Transactions on Knowledge and Data Engineering* 34.1 (2020), pp. 50–70.

[14] Abdalraouf Hassan and Ausif Mahmood. "Deep learning for sentence classification". In: *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. 2017, pp. 1–5. DOI: 10.1109/LISAT.2017.8001979.

[15] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[16] Joel Nothman et al. "Learning multilingual named entity recognition from Wikipedia". In: *Artificial Intelligence* 194 (2012), pp. 151–175. DOI: 10.1016/j.artint.2012.03.006. URL: http://dx.doi.org/10.1016/j.artint.2012.03.006.

[17] Bryan Wilie et al. "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. 2020.

[18] Alan Ritter et al. "Named Entity Recognition in Tweets: An Experimental Study". In: *EMNLP*. 2011.

[19] Maarten Sap et al. "Social Bias Frames: Reasoning about Social and Power Implications of Language". In: *ACL*. 2020.

[20] Lucy Lu Wang et al. "Cord-19: The covid-19 open research dataset". In: *ArXiv* (2020).

[21] OpenAI. *GPT-3.5 language model*. https://openai.com/. Retrieved April 25, 2023. 2021.