Pipeline for the Automatic Extraction of Procedural Knowledge from Assembly Instructions into Controlled Natural Language

Christine Rese[1, 2*], Nikolai West[2], Mathias Gebler[1], Sven Krzoska[1, 2], Philipp Schlunder[3], Jochen Deuse[3, 4]

[1] Volkswagen AG, Berliner Ring 2, 38436 Wolfsburg, Germany.
[2] Institute for Production Systems, Leonhard-Euler-Str. 5, 44227 Dortmund, Germany.
[3] RapidMiner GmbH, Westfalendamm 87, 44141 Dortmund, Germany.
[4] Centre for Advanced Manufacturing, University of Technology Sydney, 15 Broadway Ultimo NSW 2007, Australia.

* Corresponding author. Email: christine.rese@volkswagen.de
Manuscript submitted October 10, 2022; revised November 8, 2022; accepted December 10, 2022.
doi: 10.17706/jsw.18.1.1-14

Abstract: This paper presents the application of a Natural Language Processing (NLP) pipeline, which automatically extracts procedural knowledge in a standardized way from assembly instructions. The developed pipeline is able to parse and process written German assembly instructions regardless of the language discourse. The pipeline helps resolve ambiguities in assembly instructions by converting them into a Controlled Natural Language (CNL). The pipeline fully automates the translation process from free-text assembly instructions to CNL representations. We investigated and evaluated the efficiency and robustness of the NLP pipeline along multiple dimensions, such as different assembly process designers, language and fuzzy string matching models. To test the developed pipeline we used to automatically extract procedural knowledge in a standardized way for 2,740 assembly instructions obtained from automotive industry. Our investigation shows that the NLP pipeline is able to extract CNL representations with high accuracy (ø 87%). Downstream applications, such as assembly line balancing, can reuse the uniformly extracted procedural knowledge.

Keywords: Controlled Natural Language, Industrial Data Science, Information Extraction, Natural Language Processing, Text Mining

1. Introduction

In order to satisfy the increasing diversity of customer demand, manufacturing companies have to manage an ever-growing number of product variants [1]. From an economic point of view, the resulting variability has to be categorized into the two groups of value-adding and non-value-adding variability. The former constitutes waste according to the philosophy of Lean Thinking [2]. Taiichi Ohno originated Lean Thinking at Toyota in 1988 [3]. Since then, practitioners strive to eliminate waste with established tools of Industrial Engineering (IE). The second type results directly from customer requests and represents an additional value through individualization, which customers seek [2]. Here, conventional methods of IE can reach their limit due to the high number of variants and the high degree of variability. During the past decade, tools and methods from the domain of Data Science began to trickle into IE, where they are nowadays referred to as applications of Industrial Data Science (IDS) [4]. The methods promise novel solutions for the increasingly diverse planning and scheduling tasks of the manufacturing industry.

Particularly in assembly lines, the total number of product variants has a significant impact on the complexity, as all variations have to be taken into account during planning and production [5]. The whole process relies on a comprehensive documentation that summarizes and sustains the assembly process in a defined data structure [6]. Hence, such textual instructions for assembly operations contain valuable process knowledge, e.g. a detailed itemization of process steps or information about hierarchical dependencies for component assembly [7]. Through the application of traditional IE methods alone, manufacturing companies are not able to exploit the full potential contained in this data [8]. The language used in assembly operations, with regard to grammar and vocabulary, can vary vastly, either from plant to plant or from user to user [9]. Additionally, free text, abbreviations and acronyms cause ambiguities and a lack of consistency. Controlled Natural Language (CNL) provides a remedy as it facilitates the reuse and repurpose of procedural knowledge [7]. Nonetheless, unstructured and non-standard assembly instructions are often prevalent and compromise the integrity of the CNL framework. To overcome the inconsistencies, IDS provides a provision solution in the form of Natural Language Processing (NLP). Methods and tools of NLP can enable a data-driven extraction and a machine-assisted understanding of free-text descriptions of assembly operations [9]. For this reason, this paper presents an approach for data-driven extraction of knowledge from written assembly instructions. Using NLP, ambiguous texts are converted in a standardized format, relating to CNL. The main goal is the development of a pipeline that applies NLP steps to automatically extract and translate information into standardized CNL representation. The processing pipeline is evaluated by using German assembly instructions of various door assembly plans from the automotive industry. Additionally, a comparison for each part of the ensemble model is evaluated.

The remaining paper is organized in six parts. First, a brief overview of the related work on the use of CNL in manufacturing as well as on automatic extraction of procedural knowledge from textual representations of assembly information is provided (Chap. 2). Then, the NLP pipeline is described, which has been chosen for an automatic extraction of procedural knowledge (Chap. 3). Next, the case study is introduced. The case study comes from the automotive industry and forms the basis of the validation (Chap. 4). Here, the focus lies on the characteristics of the given assembly instructions and the necessary preprocessing steps. It is noteworthy that work plans from automotive industry are available for the case study. In the next chapter (Chap. 5), the introduced pipeline is demonstrated and evaluated using the case study for a group of assembly tasks. After the validation, an extensive overview of further applications for the pipeline.

2. Related Work

2.1. Controlled Natural Language for Manufacturing

The aforementioned term CNL describes subsets of natural language that can be accurately and efficiently processed by a computer, while also being expressive enough to allow natural usage by non-specialized human operators [10]. Within the manufacturing domain, a number of CNL exist, such as ASD Simplified Technical English (ASD-STE) or the Standard Language (SLANG).

ASD-STE is a CNL that originated in the aerospace industry. It is based on English and uses restrictions through about 60 general rules to make texts easier to understand, particularly for non-native speakers. ASD-STE relies on a defined vocabulary of terms common in the aerospace industry and allows the introduction of user-defined terms [11]. However, even though it is considerably more precise than full English, it does not yet allow reliable and automatic interpretation [12]. The second example is SLANG, a language developed at the Ford Motor Company that serves to create process sheets with build instructions for component and vehicle assembly plans. All SLANG instructions use an imperative mood and follow a standardized pattern starting with a main verb followed by a noun phrase. Thus, it allows writing clear and concise instructions that are also readable for a machine. At Ford, SLANG helps to automatically generate a list of required

elements and to calculate labor times [13]. These two examples serve to outline the overall extent and goal of existing models for CNL. For an in-depth description, we refer to [12] who provide a full survey and classification of existing CNL.

The Uniform Process Description (UPD) is a language for describing operations in a standardized way. The Volkswagen AG uses UPD to identify the same assembly scope of different models and variants [14]. UPD descriptions always refer to a single built-in part and just one activity. Additionally, the language can include optional information regarding the current part or component that another element in the assembly process is mounted to. In addition to the system-readable elements, text equivalents and synonyms are available in different languages. At Volkswagen, UPD is a central component for solving the Assembly Line Balancing Problem (ALBP). For this purpose, UPD describes the respective order of priority of assembly operations in the form of a precedence graph.

2.2. Automatic Extraction of Procedural Knowledge

Current research utilizes a wide range of different NLP techniques to extract procedural knowledge in an automated manner. The following segment elaborates cases of where and how NLP techniques have found application in the manufacturing domain.

Ford's Direct Labor Management System (DLMS) is a prominent example. It applies NLP to evaluate free-text part descriptions within assembly work instructions [13, 15–17]. The DLMS system uses an automotive ontology to assist in Part-Of-Speech (POS) tagging and to identify free-text part descriptions. It converts part descriptions into a standardized format and uses them for ergonomic studies. However, most research includes no in-depth description about the used NLP system and the developed ontology. In another scenario, Renu, R. S. et al. [18] apply the Stanford POS tagger to extract objects and verbs from assembly work instructions for the automatic selection of movements from a predetermined motion time system. Here, Methods-Time Measurement (MTM) is used as a basis. Based on the results of the POS tagger the automated method converts the extracted verbs into standard language and distinguishes objectives between one of five potential types: Part, consumable, tool, fixture or plant item. The combination of verb and objective type allows an automated selection from predefined tables of MTM tasks. The requirements and performance of POS taggers are investigated using assembly work instructions [9]. This research presents the Stanford CoreNLP as the best approach to parse assembly process information in text. However, the research takes data of only a single company into account. Additionally, Costa, C. M. et al. [19] present an approach that applied the Stanford Named Entity Recognizer (NER) for extracting assembly information from instruction manuals. The research shows that the Stanford NER achieves a high accuracy when identifying named entities. The model also aims to classify entities in pre-defined categories and achieves a precision of almost 90% and a F1 of around 85%. Lastly, Chen, J., & Jia, X. [20] apply an approach to discover assembly process case for assembly process reuse using multimedia information source. Here, joint application of POS and NER manages to achieve an improvement of the assembly process design.

2.3. Research Gap and Objective

This paper focuses on CNL that provide reliable and automatic semantic analysis of written assembly instructions. Related work usually assumes raw data that adheres to the format of a CNL. Furthermore, several methods and tools of NLP applications exist that can help to gain process knowledge from texts that were standardized using a CNL. As such, existing research tends to skip the uniform extraction of procedural knowledge from assembly process information. For this task, we propose a machine-learning pipeline that combines different NLP techniques and attempts to convert assembly instructions written in natural language into a CNL representation.

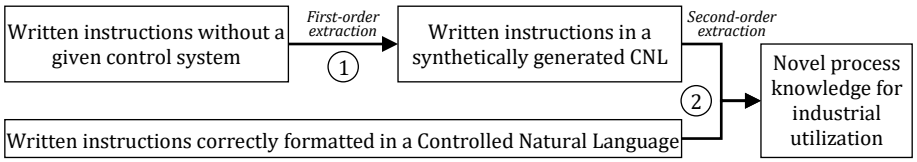

Fig. 1. Procedure to use written assembly instructions in a correctly formatted Controlled Natural Language with instructions without a control system using Natural Language Processing.

Fig. 1. summarizes the process for knowledge extraction. We refer to the extraction from natural and uncontrolled text documents in a CNL form as a first-order extraction. The extraction of process knowledge for later use then represents a second-order extraction. The goal of this step is to generalize human-written texts in the standardized format of a CNL. During a second-order extraction, both synthetically generated CNL from the first step and instructions in a CNL correctly contributed by human operators, find application. As such, both steps rely on the usage of NLP, but we consider a first-order extraction as an important preprocessing step.

3. Pipeline for Knowledge Extraction

The proposed pipeline for knowledge extraction contains three separate steps. It combines several NLP techniques to translate individual language of respective assembly process designers into a CNL representation. We outline the pipeline in Fig. 2.

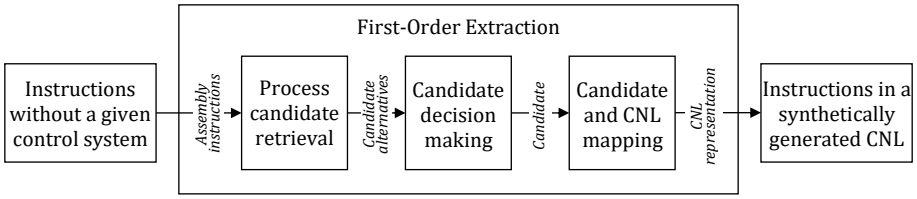

Fig. 2. Schematic structure of the NLP pipeline for performing a first-order extraction.

The pipeline relies on NLP, since it provides a wide range of methods that enable the automated processing of natural language [21]. Furthermore, it allows a robust parsing of information from written instructions of the assembly processes [9]. The focus of the pipeline lies on the execution of a first-order extraction, as described earlier.

The pipeline is useful for avoiding the time-intensive process of manually establishing a CNL. It also allows the utilization of already available process descriptions that do not yet adhere to the format of a CNL. Thus, the pipeline helps a process designer in accessing data irrespective of the data veracity and volume. The three-step approach intends to enable the translation of natural language to CNL: Process candidate retrieval, candidate decision making, and a candidate and CNL representation mapping. In the next sections, we elaborate on the tasks and goals of these phases.

3.1. Process Candidate Retrieval (Step 1)

Written assembly instructions usually contain all information required for CNL. For this reason, assembly instructions form the input of the pipeline. However, the linguistic complexity of the textual descriptions must first be reduced, e.g., by replacing tokens with regular expressions. Subsequently, the candidate retrieval process can be performed.

For the *direct comparison*, the pipeline compares assembly instructions with standard vocabulary as well as their known synonyms and abbreviations. For this reason, the NLP pipeline needs a targeted standard language. During the following validation, we will consider the Volkswagen UPD proposed by [14] as the

standard language. For this purpose, we extended the UPD with frequent synonyms and abbreviations. The pipeline performs the *direct comparison* using the Stanford Named Entity Recognizer (NER) and conducts an exact string matching. Since assembly instructions may contain more than one part, the *direct comparison* approach requires a distinction of such parts. However, the *direct comparison* ignores existing pattern in the textual descriptions. POS-Tagging can determine lexical categories like nouns, verbs, adverbs and adjectives in an assembly instruction. We use this information to differentiate between *built-in* and *add-on part* by considering existing prepositions. Prepositions are usually placed before their reference word, i.e. before the *add-on part*. The Stanford POS Tagger assigns the POS tag for each token in an assembly instruction. That helps to identify only the *built-in part* of an assembly instruction. Although there is a large number of standard vocabulary, synonyms and abbreviations, the pipeline cannot consider all of the existing designations in advance.

The second approach to retrieve potential candidates requires a further analysis of the *meaning and structure* of the assembly instructions. For this purpose, we rely on SpaCy, an effective library for advanced NLP in Python. Among other things, it allows to continue training NER model for extracting information from text. The trained NER Model identifies specific entity-based tags out of the initial data. These entities relates to the considered CNL representations (*built-in part*, *installation position*, *activity*). Therefore, we created a training corpus with 134 assembly instructions. The custom NER enables also the identification of those designations that are not found by the *direct comparison* approach (e.g. misspelled words).

Another approach extracts procedural knowledge from stored part numbers. The instruction level usually contains information about the part number. However, assembly process designers not always maintain part numbers at instruction level and moreover, part numbers are not always clearly assigned to a specific part.

3.2. Candidate Decision Making (Step 2)

The second step of the NLP pipeline is to select a candidate for each text category of the CNL. The different outlined extraction approaches can result in different recommended candidates for the CNL representations, such as *built-in part*, *activity* or *installation position*. The performance of each approach is relevant to candidate decision making because it defines the order of approaches in the NLP pipeline.

In some cases, however, the candidate decision making is difficult. The reasons for these difficulties can have various causes, e.g. inaccurate or non-specific assembly instructions. Moreover, the cooperation of different process planners during the creation of a process plan can lead to changing styles or different levels of detail in the assembly instructions. Furthermore, the integration of more than one *built-in part*, multiple *activities* or different *installation positions* in a single assembly instruction can cause problems for the identification of the correct candidate. This irregular usage of mixed assembly information does not comply with today's rules for process plans. The NLP.

3.3. Candidate and CNL Representation Mapping (Step 3)

The third step converts the selected candidates into their corresponding standard language word and its numerical representation. The synthetically generated CNL is the input to the downstream applications (Chap. 4). The pipeline performs this conversion using different NLP techniques.

First, the pipeline compares each extracted text element to the standard language list using an exact matching method. Then, the pipeline compares the remaining text elements to the synonyms and abbreviations of the standard language using the same method. This order reduces ambiguity, since different standard language words can have the same synonyms and abbreviations.

Only if this does not result in a CNL representation, the pipeline considers a fuzzy matching of the extracted text descriptions and the standard vocabulary lists including synonyms and abbreviation. Fuzzy matching of the textual descriptions is quintessential to avoid only straight comparisons. Fuzzy matching models enables

the conversion of those designations that are not found by the exact matching method (e.g. misspelled words). We propose the following three fuzzy matching algorithms that help to map a candidate with the correct CNL representation: Phonetic encoding, substring matching and the Levenshtein distance. This method selection has proven itself in our application, but it may need to be supplemented according to the analysis scenario.

For the application of the algorithms, the pipeline differentiates according to the length of the extracted descriptions. The pipeline performs only substring matching for short tokens. Because both the phonetic encoding and the Levensthein distance lead for those elements to many incorrect results. For long tokens, the mapping is conducted based on the three fuzzy matching methods. However, only the candidate with the highest similarity is used. The determined similarity must be above 90%.

Table 1. Excerpt of candidates and their CNL representation including a similarity index

Candidate	CNL representation	Similarity
Tiefton-Lautsprecher *woo-fer*	Tieftonlautsprecher *woofer*	0.95
Gehäuse vollständig *housing complete*	Gehäuse (vollständig) *housing (complete)*	0.91
Lautprecher Speker	Lautsprecher speaker	0.92
Tuersteuergraet *doôr control unit*	Türsteuergerät *door control unit*	1.00
Links vorne *front lefte*	Links vorn *front left*	0.91

Table 1 shows an excerpt of candidates, where an exact matching of the textual descriptions does not lead to the corresponding CNL representation. The use of the fuzzy matching methods allows for such cases the mapping to their corresponding CNL representation. Therefore, the outlined fuzzy matching algorithms indicates the similarity of the textual descriptions.

4. Case Study

For this research, assembly instructions from automotive industry are used. Our dataset is composed of seven German historical door assembly plans of thirteen different products. Each door assembly plan contains multiple assembly instructions. These assembly instructions do not follow the CNL framework proposed by [14] yet. As such, those work instructions often contain words that are not part of the standard vocabulary. However, we chose to select only value-adding assembly instructions for the analysis. Non-value added assembly instructions are for example logistics tasks or wasted motions, like walking to get a material. By removing those non-value added assembly instructions, we ensure that only part-related assembly instructions are considered. Therefore, every assembly instruction includes at least one element of the CNL categories *built-in part* and *activity*. After preprocessing, the dataset contains 2,740 assembly instructions, which represents approx. 60% of the initial dataset. We use these instructions as input for the NLP pipeline.

Additionally, we reduce the linguistic complexity of the textual descriptions by replacing tokens with regular expressions, e.g. for range information or separators (see Table 2). Language discourse varies from

highly professional and structured to informal language. To preserve the integrity of the dataset, we do not correct exist-ing spelling mistakes and capitalization errors manually."

Table 2. Example of Initial and Preprocessed Instructions.

Initial instructions	Preprocessed instructions
door front right: assemble door handle mount bracket	door front right <SEP> assemble door handle mount bracket
plug in door control unit 2x to door harness - front left	plug in door control unit <MULT> to door harness <SEP> front left
L0L: clip clutch carrier in inner door panel	<GUELT> <SEP> clip clutch carrier in inner door panel

In addition to assembly line plan data, we also use the master data lists of USD with information of the presented CNL categories. For each category, the master data contains standard vocabulary and their numeric representations. However, the standard vocabulary and the human natural language often do not use the same terminology. Therefore, some descriptions refer to the same element, but the assembly process designers uses a synonym rather than the exact standard vocabulary. A straight comparison will not match these descriptions that is why the master data lists are extended with synonyms and abbreviations. Thus, standard vocabulary and its synonyms belong to the same numeric representation. For instance, the *activities* 'press' and 'push' are assigned to the same number. We summarize detailed information regarding the respective categories in Table 3.

Additionally, most part numbers are linked to a standard part vocabulary. Such data may help to identify the part of an assembly instruction. However, this requires the process designer to store the corresponding part number for each assembly operation.

Table 3. Overview Standard Language Master Data List

Reference data	Unique entries	Total entries
Part	1,359	5,236
Installation position	93	127
Activity	127	210

5. Validation

We apply the developed pipeline using data of the case study. Moreover, we manually label the initial dataset so that for each assembly instruction the candidate (*built-in part*, *activity* and *installation position*) as well as their CNL representations are stored. We apply and evaluate the methods and the NLP pipeline before introducing two promising use cases for the industrial utilization of the pipelines' results.

5.1. Application and Performance of Information Retrieval

In order to develop the pipeline, we first validated the performance of the outlined information retrieval approaches. For this purpose, we determined the number of extracted textual descriptions per category and compared them to the manually labeled candidates. Considering the best performance, the different approaches are then combined into the proposed NLP pipeline.

Stanford NER. Table 4 displays the performance of the *direct comparison* approach. Out of the 2,740 *activities* contained in the dataset, the *direct comparison* approach extracted 2,523 candidates. Therefore, we considered only those 92.1% of the assembly instructions to determine the accuracy of the *direct comparison* approach. From these 2,523 extracted *activities*, 95.0% correspond to the manually labeled *activities*. The remaining assembly instruction do not have an extracted *activity* candidate or the candidate is not equal to

the manual label.

Furthermore, the *direct comparison* approach determines an *installation position* for 2,182 instructions. This corresponds to 87.9% of the instructions that actually contain an *installation position*. Unlike the other categories, not every assembly instruction contains an *installation position*.

Moreover, the *direct comparison* approach identifies various instructions that contain a *built-in part* (77.6%). However, only a few of those extracted *built-in parts* are equal to the manual labels (56.7%).

Overall, the *direct comparison* approach achieves good results for the CNL categories *activity* and *installation position*. The approach achieves good results regarding those categories because they have only a limited number of synonyms and abbreviations. Errors occur mainly with neologisms or designations that are not included in the extended standard vocabulary list. In terms of the *built-in parts*, the pipeline leads to various incorrect candidates. This happens especially because the *direct comparison* approach ignores existing structures in the textual descriptions.

SpaCy Custom NER. Table 5 displays the performance of the Custom NER model. Compared against the results of the Stanford NER the Custom NER achieved results that are more accurate according to the CNL categories activity and *built-in part*. The Custom NER tagged 97.7% of the extracted *activity* candidates and 75.4% of the *built-in parts* correctly.

According to the *installation position*, 88.4% of the 2,473 identified candidates are equal to the manually labeled installation positions. Thus, the results are less accurate compared to the results of the Stanford NER.

Overall, the results show that the latter two methods already provide accurate results in terms of the categories *activity* and *installation position*. However, further procedural knowledge is quintessential to extract *built-in parts* correctly. For this reason, the pipeline additionally takes into account information about the part number.

Table 4. Results Stanford NER

Reference data	Instructions with extracted candidates	Correctly labeled candidates
Built-in part	77.6% (2,127 of 2,740)	56.7% (1,207 of 2,127)
Activity	92.1% (2,523 of 2,740)	95.0% (2,397 of 2,523)
Installation Position	79.6% (2,182 of 2,740)	96.9% (2,114 of 2,182)
Installation Position*	87.9% (2,182 of 2,483)	96.9% (2,114 of 2,182)

*Only 2,483 of the 2,740 assembly instructions contain information about the *installation position*.

Table 5. Results Custom NER

CNL categories	Instructions with extracted candidates	Correctly labeled candidates
Built-in part	96.0% (2,630 of 2,740)	75.4% (1,983 of 2,630)
Activity	99.6% (2,730 of 2,740)	97.7% (2,666 of 2,730)
Installation Position	90.3% (2,473 of 2,740)	88.4% (2,185 of 2,473)
Installation Position*	99.4% (2,468 of 2,483)	88.5% (2,185 of 2,468)

*Only 2,483 of the 2,740 assembly instructions contain information about the *installation position*.

Part Number Approach. The part number contains information about the *built-in part* of an assembly instruction. Table 6 contains the results of the part number approach. Only about half of the assembly

instructions include information for the part number. Of those stored part numbers, 69.8% could be assigned to their respective CNL representation. From these 1,021 extracted *built-in parts*, the part number approach tags 89.1% correctly.

Table 6. Results part number approach

Example	Entries
Instructions with part number	53.4% (1,462 of 2,740)
Instructions mapped to CNL representation	69,8% (1,021 of 1,462)
Correctly mapped candidates	89.1% (910 of 1,021)

5.2. Application and Performance of NLP Pipeline

The consideration of the information retrieval approaches in the pipeline differs with respect to the CNL category. The determined performance of each approach defines their order and is thus relevant to the candidate decision making. Afterwards, the pipeline assigns the identified and selected candidates to their respective CNL representation applying exact or fuzzy matching methods.

- **Activity.** Related to the CNL category *activity*, the NLP pipeline considers first the results of the Custom NER approach. If this approach does not lead to a candidate, the pipeline considers the results of the Stanford NER.
- **Installation Position.** For the *installation positions*, the candidate decision is initially based on the Stanford NER results. The pipeline considers the Custom NER results only subsequently.
- **Built-in part.** For the *built-in part*, the information retrieval takes place via the stored part number. If no part number exists, the pipeline extends the scope of the search by taking into account the context of the assembly instruction. We realized this by creating a dictionary containing already found upstream part numbers with the found textual descriptions from the Custom NER Model. This is possible because an investigation shows that assembly process designers use similar terminology in their process descriptions for similar *built-in parts*. Only if no CNL representation is found in this way, the results of Custom NER and the Stanford NER are taken into account.

Table 7 includes the results of the pipeline for the information retrieval and the candidate decision making. For this purpose, we determined the number of extracted textual descriptions per category and compared them to the manually labeled candidates. Out of the 2,740 assembly instructions contained in the dataset, the pipeline extracted and selected more than 90% of the CNL candidates in terms of all categories. Moreover, more than 90% of those candidates correspond to the manual labels. However, the latter cannot be checked for the *built-in parts*. A comparison to the manual labels is not possible since the textual descriptions resulting from the part number approach are not identical to the manual labels. Overall, the results show that the combination of the different approaches leads to better results compared to individual methods. This applies in particular to the categories *installation position* and *built-in part*.

Table 8 presents the results of the candidate and CNL representation mapping. For this purpose, we determine the number of the instructions with a mapped CNL category and compare them to the manually inserted CNL representations. For 95.4% of the assembly instructions the pipeline mapped a CNL

representation for the category *activity*. Of those 2,613 extracted CNL representations 100% are mapped correctly. The CNL representations of the installation positions are also almost completely correct. However, the number of instructions with an identified CNL representation is lower. In terms of the *built-in parts*, the candidate and CNL representation mapping works for 74.5%. Of those, 2,040 CNL representations 81.4% are mapped correctly.

In some cases, assignment to a specific CNL representation is still a problem. Possible reasons for this problem are e.g., inaccurate or non-specific assembly instructions as well as the multiple use of same synonyms for different CNL representations. Another influencing factor is incorrect data. In addition to spelling mistakes or the use of neologisms, missing or multiple statements of different part numbers within an assembly instruction are other reasons for difficulties in the CNL candidate mapping. Due to their specific characteristics, different influences in the creation and maintenance of process plans lead to a number of challenges during the identification of candidates and mapping them to the respective CNL representations.

Table 7. Pipeline results information retrieval and candidate decision-making

CNL categories	Instructions with extracted candidates	Correctly labeled candidates
Built-in part	99.6% (2,730 of 2,740)	-**
Activity	99.7% (2,732 of 2,740)	97.6% (2,668 of 2,732)
Installation Position	90.6% (2,483 of 2,740)	95.0% (2,358 of 2,483)
Installation Position*	99.8% (2,478 of 2,483)	95.1% (2,358 of 2,478)

*Only 2,483 of the 2,740 assembly instructions contain information about the *installation position*.

**Candidates out of the part number approach conatin standard language vocabulary which are not equal to the manually labeld candidates out of the assembly instructions.

Table 8. Pipeline results candidate CNL mapping

CNL categories	Instructions with mapped CNL representations	Correctly mapped CNL representations
Built-in part	74.5% (2,040 of 2,740)	81.4% (1,661 of 2,040)
Activity	95.4% (2,613 of 2,740)	100% (2,613 of 2,613)
Installation Position	77.2% (2,116 of 2,740)	98.3% (2,079 of 2,116)
Installation Position*	85.2% (2,115 of 2,483	98.3% (2,079 of 2,115)

*Only 2,483 of the 2,740 assembly instructions contain information about the *installation position*.

6. Further Application

We developed the pipeline as a configurable analysis module. This allows a simplified usage of the pipeline results for different downstream applications. This chapter provides a comprehensive overview of two applications that are already using the analysis module or plan to use it. The input to the downstream applications are the determined synthetically generated CNL instructions. The generated novel process

knowledge represents the second-order extraction.

6.1. Use Case 1: Line Balancing

A variety of downstream applications can benefit immensely from the information extraction and standardization. CNL representations of assembly information allows for example the creation of a product precedence graph needed for solving the assembly line balancing problem in practice [14]. For this purpose, Gebler, M. [14] uses a graph generation approach, which creates a sufficient precedence graph from feasible production sequences [22]. The graph generation approach uses former production plans available in the firm to collect precedence relations between operations. From this a so called maximum graph with proven independences and not proven precedence relations is built by examining all possible combinations of two operations in the sequences. If an operation i is in each sequence executed before j than i is a predecessor of j. However, if one sequence exists where i is conducted before j and another sequence exists where i is executed after j than i and j are independent.

The basic concept of generating a maximum graph has some disadvantages [23]. The benefit highly depends on the characteristics of the input sequences. Degrees of freedom can only be deduced from pairs of operations where both possible orders were planned. For this reason, the approach achieves its effectiveness only months after the start of production when different production plans are available. Even then, it is not guaranteed that certain degrees of freedom will be found because some independent operations never change their relative position to each other.

Using the results of the first-order extraction can solve one of the significant disadvantages of the basic concept. For this purpose, the CNL *built-in part* representation is used as a preprocessing step before generating the sequences out of the production plans. After that, the CNL *built-in part* representation serves for building up the sequences instead of the operation identification. These can then be used to create a product maximum graph [14]. The advantage of this approach is that sequences with standardized elements are used. For this reason, production plans from the previous model as well as production plans from other assembly lines with similar or identical products can be used.

6.2. Use Case 2: Data-based Decision Support for Work Process Design and Optimization

The process design and optimization task in manufacturing can also benefit immensely from the information extraction and standardization. Currently, design and optimization solutions for manual processes in manufacturing are highly dependent on domain knowledge and individual experience. With an increasing number of products and a large production network, the personal network and previous experiences lose importance. At the same time, establishing cross-connections and transferring knowledge is a challenging task.

In terms of the design and optimization task, the results of the first-order extraction reduces the manual effort required to identify relevant data. Synthetically generated CNL allows for example the automatic identification of best practices in manufacturing processes. This gives assembly process designers the ability to fulfill their design and optimization tasks more efficiently and effectively. We use the machine-readable assembly information to develop a data-driven decision-making based on a three-step approach.

- Step 1: Developing a methodology to identify similar work systems
- Step 2: Developing an automated benchmarking, which compares and evaluates similar work systems for work process design and optimization tasks
- Step 3: Developing and implementing data mining methods for the contextual provision of proposals for work process design and optimization

In this application, synthetically generated CNL allows comparisons of different part-related process sequences regardless of language usage and of the language used in the textual descriptions. Overall, the machine-readable assembly instructions lead to proactive, databased-decision making and reduce time and effort expended in process design and optimization tasks.

7. Conclusion

This paper contributes to the research on automatic acquisition of procedural knowledge from textual description and CNL. We propose a novel NLP pipeline that enables the first-order extraction of procedural knowledge and retrieval of synthetically generated CNL. Our research showed that an automatic translation of assembly instructions into CNL is possible mostly regardless of the language discourse.

For the first-order extraction, we use information from assembly instructions and part numbers. Both are stored at operation level. By implementing and merging different NLP techniques, we obtained the procedural knowledge (*built-in part, installation position, activity*) which is then translated into a machine-readable format (synthetically generated CNL).

The presented knowledge extraction pipeline has proven itself in a use case, as shown in the validation. To ensure universal applicability of the pipeline, further applications in other areas and domains are to be carried out. Of particular interest are synergies in the extraction of similar text documents from different application areas. This may lead to a joint analysis of documents across the entire value stream, e.g. from product design, process planning and quality management. Due to the high proximity of the pipeline to the presented use case, further work is required especially for a generally valid abstraction.

In future, we plan to use the NLP pipeline for the whole assembly plan, other languages, and crafts. Moreover, we plan to evaluate the second-order extraction based on the results of the first-order extraction in terms of the outlined applications.

Acknowledgment

The work on this paper has been supported by the German Federal Ministry of Education and Research (BMBF) as part of the funding program 'Industry 4.0 - Collaborations in Dynamic Value Networks (InKoWe)' in the research project AKKORD (02P17D210); *www.akkord-projekt.de/en*.

References

[1] Richter, R., Deuse, J., Willats, P., Syberg, M., & Lenze, D. (2021). Managing Variability in Production. *IFIP Advances in Information and Communication Technology, Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G. & Romero, D. (Eds.), Cham: Springer International Publishing, 730–738.

[2] Deuse, J., *et al.* (2016). Pushing the Limits of Lean Thinking: Design and Management of Complex Production Systems, *Closing the Gap Between Practice and Research in Industrial Engineering*, San Sebastián, Spanien, 335–342.

[3] Ohno, T. (2008). Toyota production system: Beyond large-scale production. 1st ed. New York, NY: Productivity Press, 2008.

[4] West, N., Gries, J., Brockmeier, C., Gobel, J. C., & Deuse, J. (2021). Towards integrated data analysis quality: criteria for the application of industrial data science, *IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*, 22(1), 131–138, 2021, doi: 10.1109/ IRI51335.2021.00024.

[5] Bley, H, & Zenner, C. (2006). Variant-oriented Assembly Planning. *Annals of the CIRP*, 55(1), 23–28, doi: 10.1016/S0007-8506(07)60358-8.

[6] Erohin, O., Kuhlang, P., Schallow, J., & Deuse, J. (2012). Intelligent utilisation of digital databases for assembly time determination in early phases of product emergence. *CIRP Conference on Manufacturing Systems*, 45(3), 424–429, doi: 10.1016/j.procir.2012.07.073.

[7] Manns, M., Wallis, R., & Deuse, J. (2015). Automatic proposal of assembly work plans with a controlled natural language, *Procedia CIRP*, 33(1), 345–350, doi: 10.1016/j.procir.2015.06.079.

[8] Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., & Barbaray, R. (2018). The industrial management of SMEs in the era of Industry 4.0, *International Journal of Production Research*, *56(3),* 1118–1136, doi: 10.1080/00207543.2017.1372647.

[9] Renu, R. S., & Mocko, G. (2015). Design and implementation of a line balance visualization and editing tool. In A. Chakrabarti (Eds.), *Smart Innovation, Systems and Technologies, ICoRD'15 – Research into Design Across Boundaries Vol. 2*, (pp. 275–289). New Delhi: Springer India.

[10] Fuchs, N. E., & Schwitter, R. (1994). Specifying logic programs in controlled natural language, *Workshop on Computational Logic for Natural Language Processing (CLNLP), Edinburgh, 1994 April 3-5*, 1–16.

[11] ASD (2021). Simplified Technical English: AeroSpace and Defence Industries Association of Europe, *Specification ASD-STE100*, Issue 8.

[12] Kuhn, T. (2014). A survey and classification of controlled natural languages. *Computational Linguistics*, 40(1), 121–170, doi: 10.1162/COLI_a_00168.

[13] Rychtyckyj, N. (2005). Ergonomic analysis for vehicle assembly using artificial intelligence. *Proceedings of the 19th National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence* (pp. 41–50). San Jose.

[14] Gebler, M. (2021). Industrialization of optimization methods for automated assembly line balancing in the automotive industry: Industrialisierung von Optimierungsmethoden zur automatisierten Fließbandabstimmung in der Automobilindustrie (Original Title). Dissertation, Friedrich Schiller University, Jena.

[15] Klampfl, E., Gusikhin, O., & Rossi, G, (2006). Optimization of workcell layouts in a mixed-model assembly line environment. *International Journal of Flexible Manufacturing Systems, 17(4),* 277–299, doi: 10.1007/s10696-006-9029-6.

[16] Rychtyckyj, N. (2007). Intelligent systems for manufacturing at ford motor company. *IEEE Intelligent Systems,* 22(1), 16–19, doi: 10.1007/BF01794633.

[17] Rychtyckyj, N., Klampfl, E., & Rossi, G. (2007). Application of Intelligent Methods to Automotive Assembly Planning. *Proceedings of 2007 IEEE International Conference on Systems*. *Man and Cybernetics* (pp. 2479–2483).

[18] Renu, R. S., Mocko, G., & Koneru, A. (2013). Use of big data and knowledge discovery to create data backbones for decision support systems. *Procedia Computer Science,* 20(1), 446–453.

[19] Costa, C. M., Veiga, G., Sousa, A., & Nunes, S. (2017). Evaluation of stanford NER for extraction of assembly information form instruction manuals. In L. Marques, & A. Bernardino (Eds), *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (pp. 302–309). Piscataway, NJ.

[20] Chen, J., & Jia, X. (2020). An approach for assembly process case discovery using multimedia information source. Computers in Industry, 15, 103176, doi: 10.1016/j.compind.2019.103176.

[21] Teufl, P., Payer, U., & Lackner, G. (2010). From NLP (Natural Language Processing) to MLP (Machine Learning Processing). In I. Kotenko, & V. Skormin (Eds.), *Computer Network Security,* (pp. 256–269). Berlin, Heidelberg: Springer.

[22] Klindworth, H., Otto, C., & Scholl, A. (2012). On a learning precedence graph concept for the automotive industry. *European Journal of Operational Research, 217(2)*, 259–269, doi: 10.1016/j.ejor.2011.09.024.

[23] Otto, C., & Otto, A. (2014). Multiple-source learning precedence graph concept for the automotive

industry. *European Journal of Operational Research, 234(1),* 253–265, doi: 10.1016/j.ejor.2013.09.034.