Artificial Intelligence for Engineering Design, Analysis and Manufacturing

cambridge.org/aie

Automated feedback generation for formal manufacturing rule extraction

SungKu Kang[1], Lalit Patil[1], Arvind Rangarajan[2], Abha Moitra[2], Tao Jia[3], Dean Robinson[2] and Debasish Dutta[4]

[1]University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA; [2]General Electric Global Research, Niskayuna, New York 12309, USA; [3]General Electric Healthcare, Waukesha, Wisconsin 53188, USA and [4]Rutgers University, New Brunswick, New Jersey 08901, USA

Research Article

Cite this article: Kang SungK, Patil L, Rangarajan A, Moitra A, Jia T, Robinson D, Dutta D (2019). Automated feedback generation for formal manufacturing rule extraction. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **33**, 289–301. https://doi.org/10.1017/S0890060419000027

Received: 26 March 2018
Revised: 22 January 2019
Accepted: 3 February 2019
First published online: 19 March 2019

Key words:
Automated feedback generation; Natural Language Processing; semantic technology; ontology

Author for correspondence:
SungKu Kang, E-mail: skang47@illinois.edu

Abstract

Manufacturing knowledge is maintained primarily in the unstructured text in industry. To facilitate the reuse of the knowledge, previous efforts have utilized Natural Language Processing (NLP) to classify manufacturing documents or to extract structured knowledge (e.g. ontology) from manufacturing text. On the other hand, extracting more complex knowledge, such as manufacturing rule, has not been feasible in a practical scenario, as standard NLP techniques cannot address the input text that needs validation. Specifically, if the input text contains the information irrelevant to the rule-definition or semantically invalid expression, standard NLP techniques cannot selectively derive precise information for the extraction of the desired formal manufacturing rule. To address the gap, we developed the feedback generation method based on Constraint-based Modeling (CBM) coupled with NLP and domain ontology, designed to support formal manufacturing rule extraction. Specifically, the developed method identifies the necessity of input text validation based on the predefined constraints and provides the relevant feedback to help the user modify the input text, so that the desired rule can be extracted. We proved the feasibility of the method by extending the previously implemented formal rule extraction framework. The effectiveness of the method is demonstrated by enabling the extraction of correct manufacturing rules from all the cases that need input text validation, about 30% of the dataset, after modifying the input text based on the feedback. We expect the feedback generation method will contribute to the adoption of semantics-based technology in the manufacturing field, by facilitating precise knowledge acquisition from manufacturing-related documents in a practical scenario.

Introduction

The most prevalent form of knowledge in the industry has been text [approximately 80% (Yu *et al.*, 2005; Ur-Rahman and Harding, 2012)]. However, the reuse of the textual knowledge has been difficult since extracting useful knowledge requires the close examination by domain experts. To address the issue, (semi-)automatically extracting useful information from unstructured text has been actively studied in the manufacturing field. Previous efforts include the classification of the design and manufacturing text (Riel and Boonyasopon, 2009; Boonyasopon *et al.*, 2011; Shotorbani *et al.*, 2016), the extraction of structured knowledge (Li *et al.*, 2005; Yang *et al.*, 2005; Li and Ramani, 2007; Choi *et al.*, 2012; Cheong *et al.*, 2016; Jeon *et al.*, 2016; Wang *et al.*, 2016), and the creation of ontology from text (Ahmed *et al.*, 2007; Li *et al.*, 2007, 2009; Ameri *et al.*, 2014) based on the identification of manufacturing-related terms such as "machining" and "casting" or the identification of simple association relations such as "A is B" and "A has B". On the other hand, the extraction of more complex knowledge, such as manufacturing rule extraction, has not been focused in the manufacturing field regardless of its usefulness (Kang *et al.*, 2015). Specifically, if formal manufacturing rule extraction becomes feasible, the manufacturing rules in documents become easily accessible from CAD applications for complex tasks including early manufacturability analysis (Rangarajan *et al.*, 2013). Specifically, while the formal manufacturing rules are not valid in every case, the formal rules can help a designer to identify possible manufacturability issues that might have been missed during the design. Then the issues can be double-checked and easily fixed before the later stage of manufacturing.

Rule extraction from text is known as a challenging task since it requires an additional modeling step for joining multiple inter-concept relations into the desired rule (Kang and Lee, 2005). Moreover, to obtain the desired rule, it is important to join the inter-concept relations in the way that the extracted rule performs the intended behavior. Therefore, if the text contains the information irrelevant to the rule-definition or a semantically invalid expression, it is necessary to identify and modify the expression to ensure the extraction of the desired manufacturing rule. For example, when a formal manufacturing rule is extracted from the

© Cambridge University Press 2019

CAMBRIDGE UNIVERSITY PRESS

sentence "sharp corners should be avoided because they interfere with metal flow", the information corresponding to the clause "they interfere with metal flow" should be marked as irrelevant, as it is not relevant to the desired behavior of the rule, "identify and avoid sharp corners". However, automatically identifying and revising such expressions is not feasible with standard Natural Language Processing (NLP) techniques, since addressing the expressions requires the understanding of highly domain-specific context.

To address the gap, this paper proposes the feedback generation method to validate input text for formal manufacturing rule extraction. Specifically, the developed method identifies the necessity of input text validation and provides the relevant feedback to help the user revise the input text for the acquisition of correct rules. To the best of our knowledge, there is no literature that discusses the feedback generation for formal manufacturing rule extraction. Therefore, we developed the method that utilizes the existing feedback generation technique, Constraint-based Modeling (CBM), coupled with standard NLP techniques and manufacturing domain ontology. It is known that standard NLP techniques, trained from a generic corpus, cannot reliably consider the context required to interpret highly domain-specific text (Yang *et al.*, 2011). Therefore, to verify the feasibility of the method, we assumed that (1) NLP techniques have acceptable performance on a domain of interest, (2) the input text is written in the relatively formal language, and (3) a domain ontology is written to cover all the concepts in the input text. Under these assumptions, we proved the feasibility of the feedback generation method by applying it to the previously developed formal manufacturing rule extraction framework (Kang *et al.*, 2015). Specifically, the effectiveness of the method is demonstrated by enabling the extraction of correct manufacturing rules from all the cases that need input text validation, which is about 30% of the dataset.

The rest of this paper is organized as follows. First, the previous works related to the subject are reviewed in "Relevant work" section. Then the problem definition is shown in "Problem definition" section, and the overview of the feedback generation framework is shown in "Overview of feedback generation method" section. From "Extract manufacturing concepts/relations" section to "Identify triggered constraint and generate feedback" section, the detail of each stage of feedback generation is explained. Lastly, the implementation of the method and the experimental result are shown in "Implementation and experiment" section, and the conclusion and future challenges are provided in "Conclusion" section.

Relevant work

In this section, the previous efforts on feedback generation are reviewed, and the reason for adopting CBM as the basis of our feedback generation method is followed.

Feedback generation

Feedback generation refers to the activity (1) identifying differences between the appropriate standards for any given work and (2) providing relevant information to generate improved work (Boud and Molloy, 2013). Since automated feedback generation has been regarded as an essential feature of Intelligent Tutoring System (ITS), especially for distance learning and online learning systems where the number of students is much larger than that of tutors, it is actively studied in education field. In the rest of this section, the brief review of the existing feedback generation methods is followed. Please note that this paper sticks to the categorization of feedback generation methods presented by Keuning *et al.* (2016). For additional details, the readers are referred to the literature (Keuning *et al.*, 2016).

The first type of feedback generation methods is Model Tracing (MT) that creates feedbacks based on the procedure performed by a user. Specifically, each step performed by a user is compared with the desired procedure, and the hint or solution is provided when the user failed to take the desired step. The LISP tutor (Corbett *et al.*, 1990) is an example of an MT system that utilizes a set of production rules to compare each step performed by a user and the desired procedure. On the other hand, CBM (Ohlsson, 1994) only considers the user's final input, rather than the historical path of the user's input. The method relies on a set of pre-defined constraints that defines (1) the satisfaction condition of the potential issue and (2) the relevant feedback to be provided when the satisfaction condition is triggered. Specifically, when the user's input triggers the satisfaction conditions of some constraints, relevant feedback is automatically provided to help the user to fix the issue. The examples of CBM-based systems include the system named DB-suite (Mitrovic *et al.*, 2004) and INCOM (Le and Pinkwart, 2011). DB-suite (Mitrovic *et al.*, 2004) has a CBM-based tutor that utilizes a knowledge base of constraints. Using the constraints, the system identifies if the user's input is fully or partially equivalent to the desired input. The system then provides the feedback defined in the triggered constraints. Similarly, INCOM (Le and Pinkwart, 2011) utilizes a semantic table, which includes the constraints with different weights, and transformation rules to identify the intention of a user and generate feedbacks. However, CBM is known to be ineffective when the solution space is large or multiple solution variants are available, since different constraints may be required for each of the solution variants (Le and Pinkwart, 2011).

Other recently introduced methods include Artificial Intelligence (AI)-based and Machine Learning (ML)-based methods. For example, Lane and VanLehn (2005) developed a dialogue-based intelligent tutoring system ProPL that mimics the conversation between a human tutor and a user. Macnish developed the system datlab (MacNish, 2002) that utilizes an Artificial Neural Network (ANN) to classify the error of a user and generate relevant feedbacks. Ferreira and Atkinson (2009) developed a feedback generation system for foreign language tutoring using a simple decision tree to choose the type of feedback to be given. Gutierrez and Atkinson, 2011 utilized the combination of two ML methods, Support Vector Machine (SVM) and Conditional Random Field (CRF), trained from a set of the actual user–tutor interactions to generate adaptive feedback for the virtual foreign language tutor. NLP has been frequently used for ITS especially for a foreign language tutor. Nagata (2002) developed a system named BANZAI that utilizes NLP to analyze the Japanese input of a user and generates the relevant feedback if the input has a grammatical deficiency. Dzikovska *et al.* (2014) developed a tutoring system for basic electricity and electronics, called BEETLE II. Their system also utilizes NLP coupled with ontology to analyze the user's input and find out if some of the required concepts are missing. The system can generate dynamic feedbacks based on the result of the analysis. Even though ML-based and ANN-based methods are potentially powerful mechanisms to generate proper feedback, they are not applicable to feedback generation for manufacturing rule extraction. First, a relatively small amount of manufacturing-related data is accessible due to confidentiality issues, which prevents the use of

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

unsupervised methods to perform decent clustering. Second, to the best of our knowledge, none of the accessible manufacturing text provides the specific context (e.g. annotation of the necessity of feedback), while actual grading data can provide the context to generate feedbacks for tutoring purpose.

Feedback generation for formal manufacturing rule extraction

As mentioned, automated feedback generation has not been studied in the manufacturing field for formal rule extraction. Thus, it is necessary to extend an existing method to perform feedback generation for manufacturing rule extraction. In this paper, CBM is adopted as the basis of our feedback generation method for the following reasons.

(1) CBM does not require standard training resources as recent AI-based or ML-based methods do.
(2) CBM can be used without inspecting the procedure of the action performed by a user while MT cannot.
(3) The cases that need input text validation, such as the inclusion of a subordinate clause with the information irrelevant to the rule-definition, can be modeled as the constraints that CBM relies on. Also, CBM is effective since the number of solution variants is small.

To adopt CBM to develop the feedback generation method for formal manufacturing rule extraction, previously extracted manufacturing rules are analyzed to model the constraints that identify the necessity of input text validation. When a constraint is triggered, the relevant feedback is provided so that the user can modify the input text to extract the correct manufacturing rule.

Problem definition

The left side of Figure 1 shows the manufacturing rule extraction from the text, "Sharp corners should be avoided because they interfere with metal flow", performed by the previously developed rule extraction framework (Kang *et al.*, 2015). Specifically, the extracted rule is written to identify the existence of *Corner* and *MetalFlow* in a given CAD model, and then notify the designer if the Corner is *isSharp* and *interfereWith* the *MetalFlow* to prevent the possible manufacturability issue due to a sharp corner. However, while the desired behavior of the rule is "identifying the existence of a sharp corner", the rule is not triggered if the CAD model does not contain the instance corresponding to *MetalFlow*, which is usually incorporated in the latter stage of design, even when the CAD model contains sharp corners to be investigated. In other words, even though the rule is semantically valid, considering additional details, such as "because they interfere with metal flow", in the rule extraction process can lead to the undesired behavior due to the redundancy of the rule. Moreover, fully automatically addressing the issue is not feasible since standard NLP tools are not capable of identifying important and optional information within a highly domain-specific text. This gap becomes the motivating example for the development of feedback generation to validate the input text for the extraction of the desired formal rule. To address the issue, the main function of the feedback generation method is to identify the necessity of input text validation and provide relevant feedback to the user. Based on the feedback, the user can modify the input text and acquire the correct rule with the rule extraction framework. For example, Figure 1 shows that the feedback suggests the user check the subordinate clause and enclose the clause with parenthesis if the information is irrelevant to the rule-definition. With the irrelevant information marked, the desired formal rule can be extracted. Moreover, as the suggested option does not require the user to learn formal structured languages such as the Web Ontology Language (OWL) (*OWL: Web Ontology Language*, 2011) or Semantic Application Design Language (SADL) (Crapo and Moitra, 2013), the user can easily mark the information irrelevant to the rule-definition and utilize the rule extraction framework to acquire the correct rule. Developing the proposed feedback generation method requires addressing the following requirements.

Ability to identify the necessity of input text validation The cases that need input text validation should be identified. The potential issues of input text include: (1) the inclusion of the information irrelevant to the rule-definition, (2) the existence of the semantically invalid semantic relation between concepts, and (3) the input sentence without an explicit subject. For example, Figure 1 shows that the subordinate clause "because they interfere with metal flow" needs not to be considered in rule extraction, as the information is potentially irrelevant to the rule-definition.

Ability to generate the relevant feedback for input text validation Once the cases that need input text validation are identified, relevant feedback should be generated to provide helpful information including: (1) the part of input text need to be examined, (2) the detailed explanation of the potential issue, and (3) the options to address the issue. It should be noted that the user should be able to address the issue without learning formal structured languages. For example, Figure 1 shows that the feedback suggests the user enclose the irrelevant part with parenthesis so that the information is ignored in the rule extraction process.

Overview of feedback generation method

Figure 2 shows the overview of the feedback generation method. If the manufacturing text requires input text validation, some of the constraints are triggered and relevant feedback is generated to help the user address the potential issue. The entire feedback generation process is divided into the following two steps: the extraction of manufacturing concepts/relations and the identification of triggered constraints/feedback generation.

Extract manufacturing concepts and relations In this step, manufacturing concepts and relations are derived. First, natural language terms/grammatical dependencies are identified utilizing standard NLP techniques. Then the natural language terms/grammatical dependencies are matched to manufacturing concepts/relations defined in the manufacturing domain ontology. This step is important since the manufacturing concepts and relations are examined to identify the necessity of input text validation.

Identify triggered constraints and generate feedbacks In this step, relevant feedback for input text validation is provided so that the user can address the issues. To identify the necessity of input text validation, that is, to find out the candidates for feedback generation, a set of constraints are designed. Specifically, if the manufacturing concepts and relations trigger some of the constraints, the feedback attached to the constraints is provided. Based on the feedback, the user can modify the input text to acquire the desired manufacturing rule.

Extract manufacturing concepts/relations

NLP techniques can be used to analyze unstructured text to derive domain-specific structured information. For example, Pinquie

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

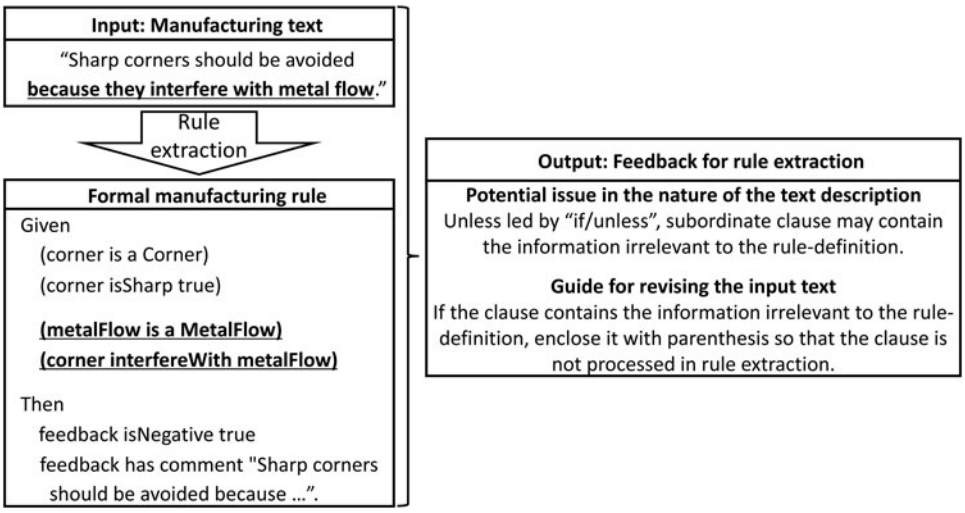


Fig. 1. The motivating example of feedback generation for the inclusion of the information irrelevant to the rule-definition (underlined) in the input text (left top) and the extracted rule (left bottom). The relevant feedback (right) is desired for the user to modify the input text for the acquisition of the correct rule.

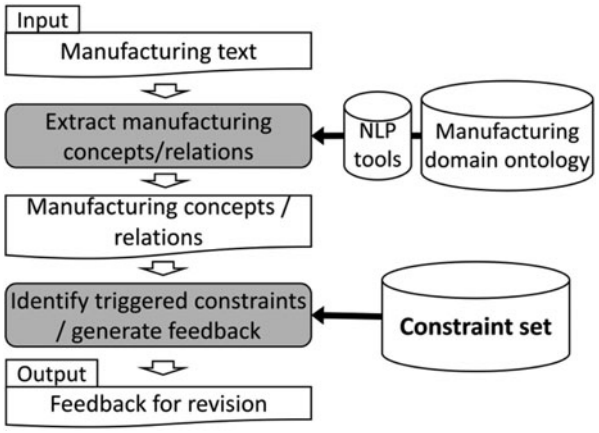


Fig. 2. The overview of the feedback generation. The block arrows indicate the sequence of execution, and the solid arrows indicate the use of tool or knowledge base.

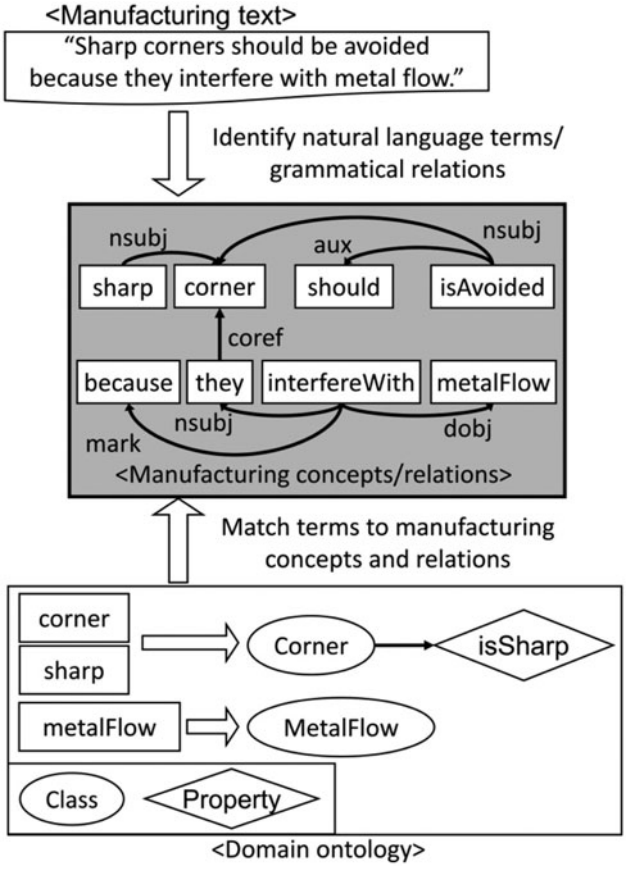


Fig. 3. The manufacturing concepts/relations derived from the motivating example. First, natural language terms/grammatical relations are identified, and then transformed to manufacturing concepts/relations defined in the manufacturing domain ontology.

et al. (2015) utilized NLP techniques, such as tokenization, lemmatization, and dependency parsing, to derive Parametric Property-Based Requirements (PPBRs) that can be imported to ENOVIA V6 for the parametric design synthesis in CATIA V6. Similarly, as the preliminary step of feedback generation, the proposed method utilizes NLP techniques to translate a manufacturing text into the manufacturing concepts and relations. The manufacturing concepts and relations are then examined to identify the necessity of input text validation. The overall process of deriving manufacturing concepts and relations is shown in Figure 3. First, standard NLP techniques are utilized to separate the manufacturing text into natural language terms and derive grammatical dependencies between the terms. Specifically, each node (e.g. "sharp", "corner", "should"…) corresponds to each term of the input text, and each connection between the nodes (e.g. "nsubj", "aux", "dobj"…) represents the grammatical dependency. In addition, co-reference resolution is performed to derive the term referred by a pronoun to resolve possible ambiguities. For example, Figure 3 shows that the pronoun "they" refers to "corners" and the co-reference relation is denoted by the dependency "coref". For the description of each type of grammatical dependency used in this paper, the readers are referred to the full list of grammatical dependencies provided in Table 1. Once the natural language terms/grammatical dependencies are

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

Table 1. The full list of grammatical dependencies used in this paper

Label	Description	Label	Description
acl	Clausal modifier of noun	mark	Marker
acomp	Adjectival complement	meta	Meta data
advcl	Adverbial clause modifier	neg	Negation modifier
advmod	Adverbial modifier	nmod	Modifier of nominal
agent	Agent (passive)	npadvmod	Noun phrase as adverbial modifier
appos	Appositional modifier	nsubj	Nominal subject
attr	Attribute	nsubjpass	Nominal subject (passive)
aux	Auxiliary verb	oprd	Object predicate
auxpass	Auxiliary verb (passive)	parataxis	Parataxis
case	Case marker	pcomp	Preposition complement
cc	Coordinating conjunction	pobj	Preposition object
ccomp	Clausal complement	poss	Possession modifier
compound	Compound word	preconj	Precorrelative conjunction
conj	Conjunct	predet	Predeterminer
coref	Co-reference	prep	Prepositional modifier
csubj	Clausal subject	prt	Verb particle
csubjpass	Clausal subject (passive)	punct	Punctuation
dative	Dative	qmod	Modifier of quantifier
dep	Unclassified dependent	relcl	Relative clause modifier
det	Determiner	root	Root
discourse	Discourse element	vocative	Vocative modifier
dobj	Direct Object	xcomp	Open clausal complement
expl	Expletive		

identified, they are transformed into the manufacturing concepts/relations defined in the manufacturing domain ontology. For example, Figure 3 shows that the terms "corner", "sharp", and "metal flow" are matched to corresponding concepts *Corner*, *isSharp*, and *MetalFlow* in the manufacturing domain ontology. It should be noted that the manufacturing concepts and relations conform to a flexible interoperable semantic model (i.e. domain ontology) rather than a proprietary standard, so the same information can be easily adapted to different semantic applications if they rely on the same upper ontology.

Identify triggered constraint and generate feedback

In this step, a set of constraints are utilized to examine the manufacturing concepts/relations so that potential issues in the nature of the input text are identified, and relevant feedback is provided. In the rest of this section, constraint check mechanism and the categories of the constraints are presented.

Constraint check mechanism

In this paper, CBM is adopted to develop the constraints that identify the necessity of input text validation and provide relevant feedback. Figure 4 shows an example of the constraints and the overview of the constraint check mechanism. A constraint has two parts: satisfaction condition and feedback. Satisfaction condition part defines the pattern of manufacturing concepts/relations that indicates the necessity of input text validation. In the example of Figure 4, the satisfaction condition is the pattern that indicates the existence of a subordinate clause in the input text. On the other hand, the feedback part defines the advice that is given to the user when the constraint is triggered. Specifically, in this paper, the form of feedback is textual advice that provides the part of the input text needs validation, the detailed explanation of the potential issue, and the options to address the issue. In the example of Figure 4, the feedback suggests enclosing the subordinate clause as it contains potentially irrelevant information. Based on the feedback, the user can address the issue by modifying the input text to acquire correct formal manufacturing rules. Figure 5 shows the extraction of the correct formal manufacturing rule after modifying the input text. Specifically, it is shown that the information irrelevant to the rule-definition, shown as the shaded region, is not included in the rule extracted from the modified sentence.

Constraint category

Based on the examination of manufacturing texts using previously developed formal manufacturing rule extraction framework (Kang *et al.*, 2015), a set of constraints are designed to identify the necessity of input text validation. The constraints are categorized into three categories based on the types of issues: (1) the constraints

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

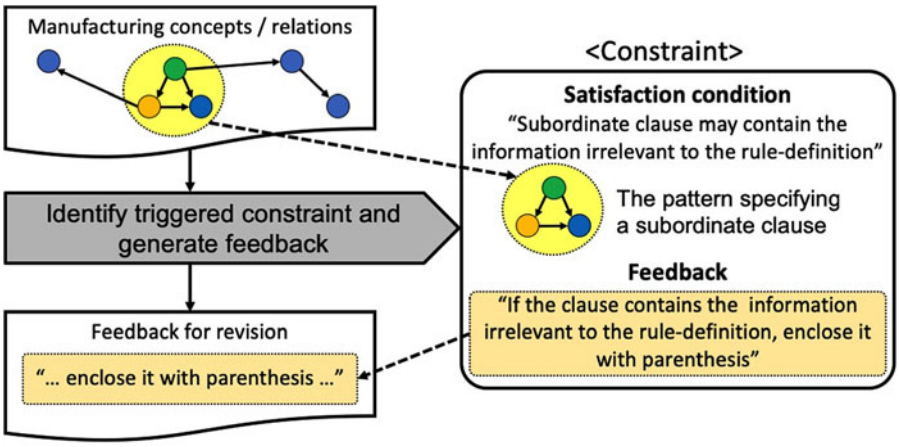


Fig. 4. An example of constraint and the overview of constraint check mechanism. The satisfaction condition of the constraint is utilized to identify if input text validation is needed, and textual feedback attached to the constraint is provided when the constraint is triggered.

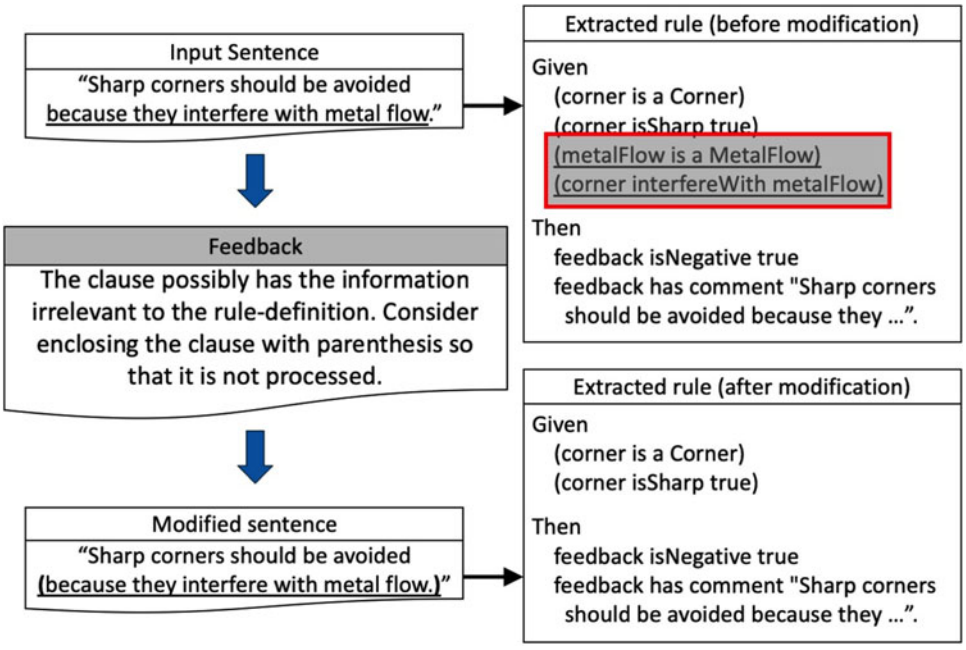


Fig. 5. Feedback generation result for the motivating example (Fig. 1). After modifying the input sentence based on the feedback, correct manufacturing rule has been extracted.

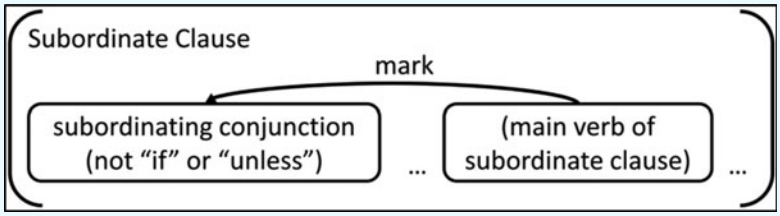
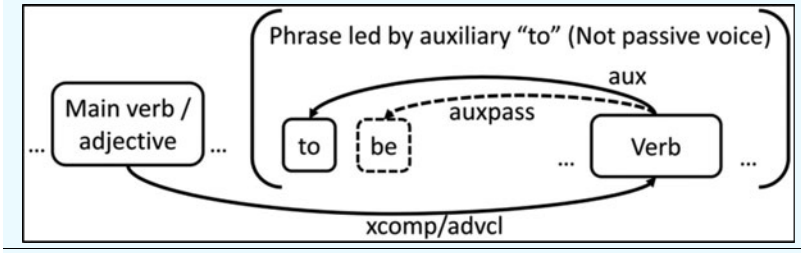
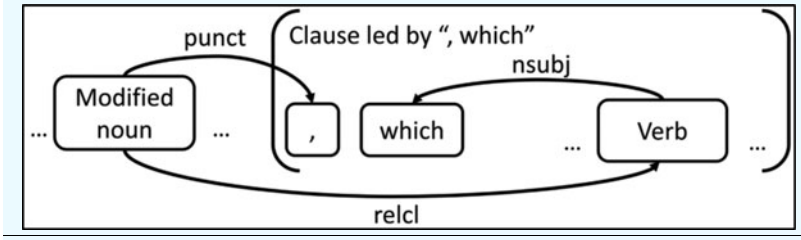
identifying the information irrelevant to the rule-definition, (2) the constraints identifying semantic disagreement, and (3) the constraints identifying sentence without an explicit subject. It should be noted that the constraints and constraint categories are independent of each other. In other words, a sentence may trigger multiple constraints in the same category or different categories.

The constraints identifying the information irrelevant to the rule-definition As shown in the motivating example (Fig. 1), a manufacturing text may contain the information irrelevant to the rule-definition, which makes the extracted rule redundant or incorrect. To address the issue, the first category of the constraints is designed to identify the existence of potentially irrelevant information. When this type of constraint is triggered, the feedback suggests the user enclose the irrelevant part with parenthesis so that the part is explicitly ignored in the rule extraction. For example, the feedback for the motivating example (Fig. 1) suggests enclosing the subordinate clause "because they interfere with metal flow" with parenthesis as it is irrelevant to the rule-definition. Table 2 shows the satisfaction conditions and the examples of the constraints in this category with the irrelevant information underlined. In the figures representing the patterns for the satisfaction conditions in Table 2, the solid line node and arrows are the manufacturing concepts/relations that should exist to trigger the constraint, and the dotted line nodes and arrows are the manufacturing concepts/relations that should not exist to trigger the constraint.

The constraints identifying semantic disagreement Sometimes, manufacturing text may represent semantically invalid or incomplete relation between two concepts, which makes the extracted rule invalid. To address the issue, the second category of the constraints is designed to identify any semantic disagreement, such as mismatching between a feature and its attributes. When this type of constraint is triggered, the feedback suggests the user modify the sentence to address the semantic disagreement based on domain context in the manufacturing domain ontology. For example, the manufacturing text "If a blind hole is required,

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

Table 2. The constraints involving the inclusion of the information irrelevant to the rule-definition (corresponding part underlined)

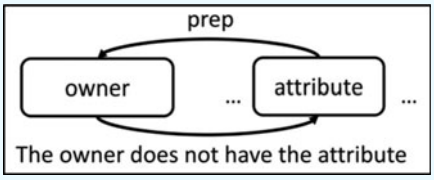
Satisfaction condition	Examples
Subordinate clause followed by a marker except for "if/unless" 	"Avoid knifelike edges of parts <u>because they interfere with the smooth flow of material through the die</u>."
Phrase involving to-infinitive (active voice) 	"<u>To facilitate removal,</u> the pattern must have taper or draft."
Non-definitive relative clause ", which" 	"Avoid sharp internal corners, <u>which concentrate heat-treating stresses</u>."
Other examples specifying the information potentially irrelevant to the rule-definition	"Avoid blind areas, <u>i.e. surfaces that are behind corners or protrusions</u>."

Solid nodes/arrows: manufacturing concepts/relations that should exist to trigger the constraint
Dotted nodes/arrows: manufacturing concepts/relations that should not exist to trigger the constraint.

its length of depth should not exceed its diameter" represents that a "blind hole" has "length of depth", but it is semantically invalid as "length of depth" is a redundant representation of "depth", which confuses rule extraction process. In this case, feedback is given to the user notifying that the attributes are not correctly related and provides the list of possible attributes of "blind hole" defined in the manufacturing domain ontology. Based on the feedback, the user can modify the attribute so that the extracted rule becomes semantically valid. Table 3 shows the satisfaction conditions and the examples of the constraints in this category with the corresponding part underlined. In the figures representing the patterns for the satisfaction conditions in Table 3, the solid line nodes and arrows are the manufacturing concepts/relations that should exist to trigger the constraint.

The constraints identifying sentence without the explicit subject Since a formal manufacturing rule consists of the triples of "subject", "predicate", and "object/value", the formal rule extracted from the input sentence without an explicit subject may contain a *null* instance. To address the issue, the third category of the constraints is designed to identify the absence of an explicit subject. When this type of constraint is triggered, the feedback suggests the user check if the actual subject is correctly inferred, and re-perform the rule extraction after re-writing the sentence with the explicit subject. For example, in the manufacturing text "It is better not to have sharp corners on a formed part", the subject of the main verb is "it", which is a dummy pronoun. Thus, the subject of the main formal statement of the extracted rule is regarded as *null* as standard NLP techniques cannot infer the actual subject. For this example, the feedback suggests re-writing the sentence with the explicit subject, such as "part" or "corner" to address the issue. Table 4 shows the satisfaction conditions and the examples of the constraints in this category with the corresponding part underlined. In the figures representing the patterns for the satisfaction conditions in Table 4, the solid line nodes and arrows are the manufacturing concepts/relations that should exist to trigger the constraint,

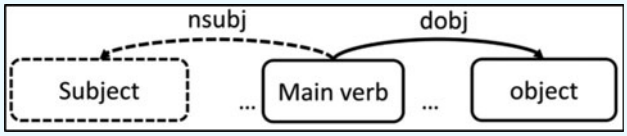
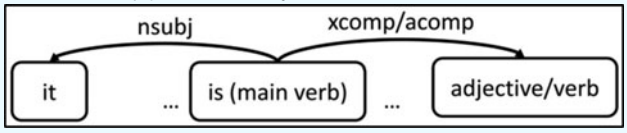
Table 3. The constraints involving semantic disagreements between the concepts (corresponding part underlined)

Satisfaction condition	Examples
Semantic disagreement between an attribute and its owner 	"If a sharp internal corner is necessary, <u>the included angle</u> should be always more than 90."

Solid nodes/arrows: manufacturing concepts/relations that should exist to trigger the constraint

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

Table 4. The constraints involving the lack of the explicit subject of the main verb (corresponding part underlined)

Satisfaction condition	Examples
Imperative mood 	"Avoid tapered-walled shells or flanged shells."
The use dummy pronoun as subject 	"It is better not to have sharp corners on a formed part."

Solid nodes/arrows: manufacturing concepts/relations that should exist to trigger the constraint
Dotted nodes/arrows: manufacturing concepts/relations that should not exist to trigger the constraint.

and the dotted line nodes and arrows are the manufacturing concepts/relations that should not exist to trigger the constraint.

Implementation and experiment

Experimental setting

Test scenario and dataset As the test scenario, the feedback generation method was applied to improve the formal manufacturing rule extraction framework previously developed by us (Kang *et al.*, 2015). For additional details about formal manufacturing rule extraction, the readers are referred to the literature (Kang *et al.*, 2015). The dataset consists of the sentences with rule-like information, chosen from the subsection "Design Recommendation" of each section of the design for manufacturing handbook written by Bralla (1998). First, the test dataset, which consists of 36 sentences in milling, metal stamping, and die-casting sections, is created to develop the method. Once the method is developed, the validation dataset is created to verify the extensibility of the method. The validation dataset contains 133 sentences from the 30 sections in forming, machining, casting, molding, assembling, and finishing chapters of the book. In creating the dataset, sentences that appeared to capture a manufacturing rule were manually selected while the sentences that appear as recommendations were discarded. The following guidelines are used to choose the sentences.

- The sentences specifying the constraint between manufacturing features are regarded as rules.
  For example, 1. "The entrance surface of a drilled hole should be perpendicular to its axis"
  For example, 2. "For steel, the depth of end-milled slots should not be deeper than the cutter diameter"
- The sentences specifying the clear preference for a design over another design are regarded as rules.
  For example, "Spotfacing is more economical than face milling for small, flat surfaces"
- The sentences specifying the recommendation without definite constraint or preference are not regarded as rules.
  For example, "The part design should be as simple as possible": Discarded since the desired degree of simple is not clear.

It should be noted that when standard NLP techniques cannot derive grammatically valid interpretation, the sentences were manually modified. That is because the purpose of this paper is to address the cases that need input text validation with respect to the semantics, but not to address grammatically invalid interpretation due to the limitation of standard NLP techniques. The following shows the guideline for sentence modification and examples (about 30% of the sentences were modified):

- Sentence structures were modified when the subject or object of the sentences could not be derived correctly.
  For example, the sentence "The minimum wall thickness for extrusions should be 0.4 mm." has been modified to "For extrusion, the minimum wall thickness should be 0.4 mm." as "extrusions" was wrongly regarded as the subject.
- Some of the terms/phrases were modified when standard NLP techniques frequently misinterpret them.
  For example, the phrase "be greater than" has been modified to "exceed" when the phrase is not correctly interpreted.
- When the sentences need too much modification, the sentences were discarded.

Furthermore, some of the sentences were manually modified such that the rules are precise to focus on domain concepts, for example, removing phrases such as "off the shelf" and replacing with the phrase "standard"; or that the rules align with the manufacturing domain ontology, for example, remove phrases such as "corner shapes" from the sentence "Slot widths, radii, chamfers, corner shapes, and overall forms should conform to …" because the manufacturing domain ontology does not capture the concept of "corner shape".

Construction of manufacturing domain ontology In this paper, manufacturing domain ontology is manually created to encode manufacturing context for the dataset. Specifically, we used SADL (Crapo and Moitra, 2013) to encode the manufacturing domain ontology. Since SADL (Crapo and Moitra, 2013) has English-like syntax, even domain experts not familiar with semantic technology can update the domain ontology when the application domain needs to be extended. SADL (Crapo and Moitra, 2013) also supports the automatic translation to standard OWL (OWL: Web Ontology Language, 2011), which is W3C recommendation built upon Resource Description Framework (RDF) (RDF 1.1 Concepts and Abstract Syntax, 2014)/Resource Description Framework Schema (RDFS) (RDF Schema, 1.1, 2014), so that the manufacturing domain ontology can be imported into semantics-based applications. Figure 6 shows the part of the manufacturing domain ontology

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

```
Corner is a type of ManufacturingFeature.
    isSharp describes Corner with a single value of type boolean.

MetalFlow is a type of MSDLPhysicalClass.
interfereWith describes Corner with values of type MetalFlow.
```

Fig. 6. A part of manufacturing domain ontology used for this paper, corresponding to the manufacturing concepts/relations in the motivating example (Fig. 1).

corresponding to the context relevant to the motivating example. In this paper, we assumed that the manufacturing domain ontology includes all the manufacturing concepts/relation mentioned in the dataset. In other words, the manufacturing domain ontology completely covers the domain of interest. In addition, the manufacturing domain ontology is not specifically designed for feedback generation process. Therefore, the feedback generation method can be extended to additional domains when the domain ontology, even designed for different purposes, is available.

Implementation

To verify the effectiveness of the feedback generation method, the previously developed Java application for formal manufacturing rule extraction (Kang *et al.*, 2015) is extended to adopt feedback generation method. The application utilizes NLP4J ("The Natural Language Processing for JVM languages (NLP4J)," n.d.) and OpenNLP ("Apache OpenNLP," n.d.) for NLP and utilizes Apache Jena Ontology API ("Apache Jena," n.d.) for the ontology management and reasoning. The application shows the procedure of feedback generation so that the user can understand how the feedback is generated. Figure 7 shows the screenshots of the application providing the relevant feedback for the motivating example. Specifically, Figure 7a shows the manufacturing concepts/relations derived from the input text, and Figure 7b shows relevant feedback to help the user validate the input text. As shown, the feedback specifies the part of the input text that needs validation and suggests the user enclose the part with parenthesis so that possibly irrelevant information can be ignored in the rule extraction process.

Result and discussion

The results of the feedback generation method are given in Figure 8. Figure 8a shows the statistics of the results that indicate how many feedbacks are created and how many true/false positives and negatives there are. First, it is shown that among 133 cases, feedback has been provided to 58 cases including 40 relevant cases (true positive), which need input text validation. Also, for all the 40 relevant cases, correct rules are extracted after modifying input sentences based on the feedback. As the constraints are designed in a conservative manner, there is no false negative, which is the case that feedback has not been provided while input text validation is needed, but feedback has been provided to 18 irrelevant cases (false positive) as well.

The results indicate that the feedback generation method can effectively help the user identify the necessity of input text validation and provide useful feedback to modify the input text for the acquisition of the desired rules. Figure 8b shows the Precision, Recall, *F*-measure of the result, which can be used as the baseline for the feedback generation for manufacturing rule extraction. In the rest of this section, several successful examples of feedback generation are shown.

Feedback generation case 1 Figure 9 shows an example of feedback generation performed by the second category of constraints. In the example, the formal rule extracted from the input sentence "If a blind hole is required, its length of depth should not exceed its diameter", shown at the top-right, was not semantically valid due to the mismatching between the feature "blind hole" and the attribute "length of depth". Specifically, the extracted rule compares "length of blind hole" with "diameter of blind hole" and regards the owner of "depth" as *null*, which is not the intended behavior. In this case, the feedback notifies that "length of depth" is not the valid attribute and suggests modifying the attribute based on the manufacturing domain ontology. Based on the feedback, the attribute is modified to "depth", and correct manufacturing rule has been extracted as shown at the bottom right of Figure 9.

Feedback generation case 2 Figure 10 shows another example of feedback generation performed by the third category of constraints. In the example, the rule extracted from the input sentence "It is better not to have sharp corners on a formed part", shown at the top-right, was not semantically valid due to the use of the dummy subject "it". Specifically, as the rule extraction framework misses the owner of "corner", extracted rule regards the owner of "corner" as *null*, which is not the intended behavior. To address the issue, the feedback notifies that the sentence does not have an explicit subject and suggests modifying the sentence with an explicit subject such as "part" or "corners". Based on the feedback, the sentence is modified to "A formed part should not have sharp corners", and correct manufacturing rule has been extracted from the modified sentence as shown at the bottom-right of Figure 10.

Conclusion

In this paper, we present the research about generating feedback to validate the input text for formal manufacturing rule extraction, so that correct manufacturing rule can be extracted after the user modifies the input text accordingly. Specifically, we demonstrated the feedback generation method that utilizes CBM coupled with standard NLP techniques and manufacturing domain ontology. We verified the feasibility of the method by extending the previously developed formal manufacturing rule extraction framework (Kang *et al.*, 2015) to incorporate feedback generation. Using the dataset created from the design for manufacturing handbook written by Bralla (1998), the effectiveness of the method is demonstrated by identifying the cases that need input text validation, which is about 30% of the dataset, and providing the relevant feedbacks to help the user. After modifying the input text based on the feedback, the extraction of the desired manufacturing rules is verified for all the cases.

To the best of our knowledge, there is no prior research focusing on the feedback generation for formal manufacturing rule extraction. We performed the research on the previous works in feedback generation and implemented the method that leverages a blend of technologies that include CBM, NLP, and semantic technology. In addition, the experimental results indicate that the feedback generation method is expected to facilitate the formal manufacturing rule extraction by helping the user identify

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

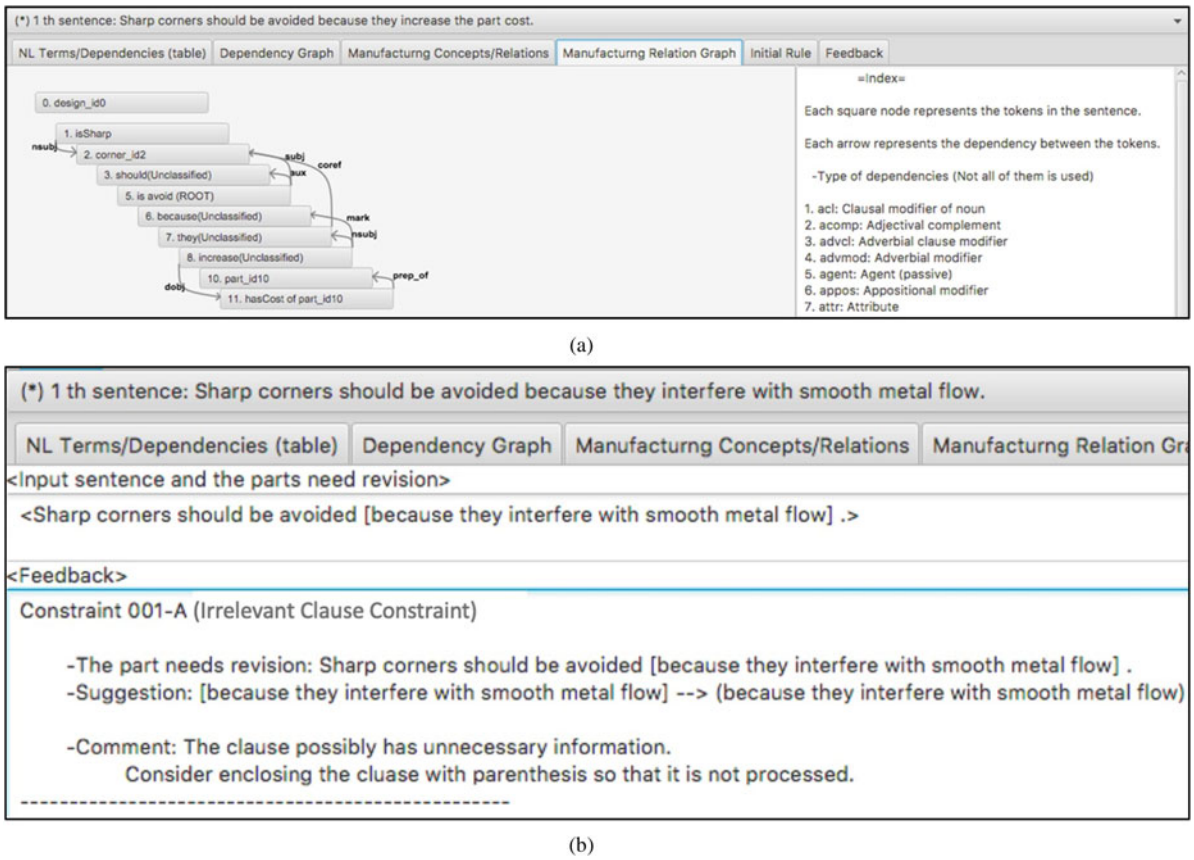


Fig. 7. The user interface of rule extraction application with feedback generation feature. (a) Manufacturing concepts and relations derived from the input text. (b) The generated feedback.

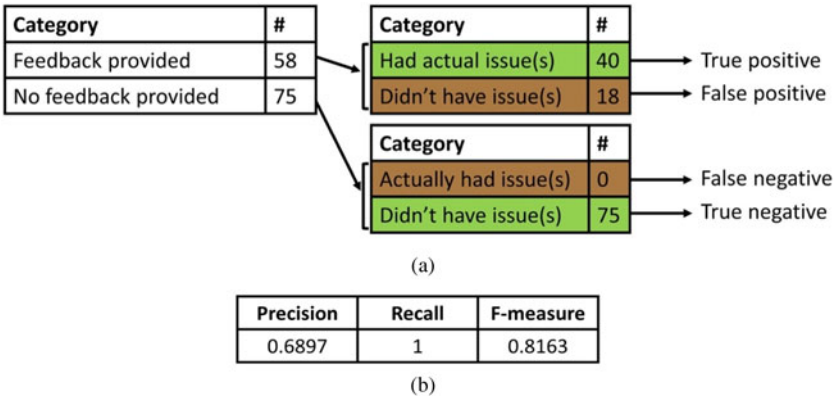


Fig. 8. The experimental results. (a) The number of the cases that feedback provided/not provided and the cases that need/do not need input text validation and (b) Precision, Recall, *F*-measure of the result.

the necessity of input text validation and providing relevant feedback to address the issue. As the proposed method is founded on the assumptions about NLP techniques, input text, and domain ontology ("Introduction" section), the cases beyond these assumptions may not be reliably handled by the proposed method. Also, due to the assumptions, currently the method requires human efforts to modify the input text and evaluate the output feedback, which may be non-trivial efforts when applied to a larger scale application. Nevertheless, our interactions with the industry partner indicate that it has a meaningful value. Specifically, it provides relevant feedback guiding the authors to write manufacturing rules accordingly, so that the textual rules can be reliably interpreted into a computer-understandable form. We also expect that the fast advance of NLP techniques will contribute to resolving the issue. Furthermore, it should be noted that the user needs not to learn a structured semantic language to address the issue since the feedback helps the user modify the input text and re-perform rule extraction.

In conclusion, we expect that the method will facilitate the development of formal manufacturing rules from unstructured text, in a practical scenario, by providing an effective mechanism to help input text validation for rule extraction. Thus, we also expect that the method can address a critical need that has prevented a larger scale application and adoption of semantic technologies in the field of manufacturing. Lastly, when domain knowledge (i.e. domain ontology) is available, the method can be

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

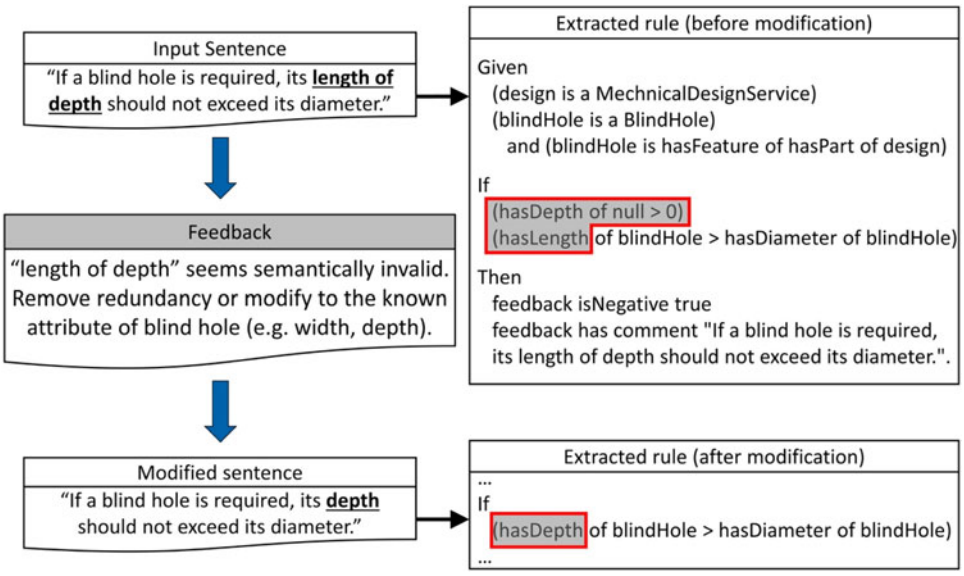


Fig. 9. The example of feedback generation. The extracted rule was not semantically valid as "length of depth" is not a valid attribute of "blind hole". After modifying the attribute to "depth" based on the feedback, the correct rule has been extracted.

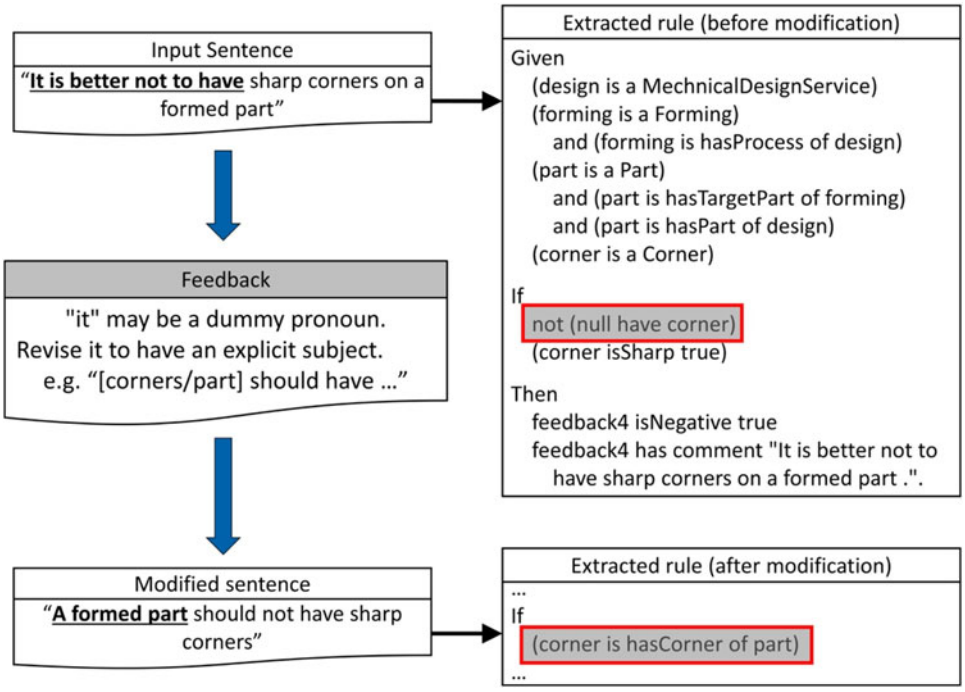


Fig. 10. The example of feedback generation for input text validation. The extract rule contained a *null* instance due to the use of a dummy subject "it". After modifying the sentence to have the subject "A formed part" based on the feedback, the correct rule has been extracted.

extended to other areas, not restricted to manufacturing-related, to inspect domain-specific text by modeling the cases that need input text validation.

Although we have made contributions to the research area of feedback generation for formal manufacturing rule extraction, much work needs to be done. The challenges we found out and future direction of the research are as follows:

The effectiveness of the feedback While the proposed method generates relevant feedbacks for potential issues in the nature of the input text, the effectiveness of the feedback depends on the user. Specifically, the user is responsible for confirming the necessity of input text validation and choosing one of the options provided by the feedback. Therefore, it is difficult to estimate the effectiveness of the feedback (i.e. how helpful is the feedback) since the same feedback may be utilized differently. In this context, there is a need for a mechanism to evaluate the effectiveness of the feedback to estimate the actual advantage provided by the feedback generation method.

The coverage of the domain ontology In this paper, we assumed that the manufacturing domain ontology covers all the

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

manufacturing-related concepts in the input dataset. However, in a practical case, some of the concepts may not be covered by available domain ontologies. To address such cases, further study is necessary to identify how to provide helpful feedback when domain ontologies partially cover the input dataset.

Further categorization of rule and recommendation In this paper, we assumed that a rule represents more definite constraints/preferences than a recommendation. However, in practice, rule, recommendation, and guideline are different in its importance and compulsion. Identifying their distinct characteristics and providing relevant feedback to help a rule author to specify them are expected to enrich the formal manufacturing rules.

Acknowledgements. Kang, Patil, and Dutta acknowledge financial support from GE Global Research. The authors would also like to thank Christine Furstoss, Global Technology Director for Manufacturing, Chemical & Materials Technologies at GE Global Research, for supporting this work.

References

Ahmed S, Kim S and Wallace KM (2007) A methodology for creating ontologies for engineering design. *Journal of Computing and Information Science in Engineering* **7**, 132.

Ameri F, Kulvatunyou B, Ivezic N and Kaikhah K (2014) Ontological conceptualization based on the SKOS. *Journal of Computing and Information Science in Engineering* **14**, 031006.

Apache Jena. Retrieved July 4, 2017, Available at https://jena.apache.org/.

Apache OpenNLP. Retrieved July 4, 2017, Available at https://opennlp.apache.org/.

Boonyasopon P, Riel A, Uys W, Louw L, Tichkiewitch S and Preez, du N (2011) Automatic knowledge extraction from manufacturing research publications. *Cirp Annals-Manufacturing Technology* **60**, 477–480.

Boud D and Molloy E (eds) (2013). *Feedback in Higher and Professional Education: Understanding it and Doing it Well*. Abingdon-on-Thames, UK: Routledge.

Bralla J (1998) *Design for Manufacturability Handbook*. New York City, USA: McGraw Hill Professional.

Cheong H, Li W and Iorio F (2016) Automated extraction of system structure knowledge from text. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V02AT03A011–V02AT03A011). American Society of Mechanical Engineers.

Choi S, Park H, Kang D, Lee JY and Kim K (2012) An SAO-based text mining approach to building a technology tree for technology planning. *Expert Systems with Applications* **39**, 11443–11455.

Corbett AT, Anderson JR and Patterson EG (1990) Student modeling and tutoring flexibility in the Lisp Intelligent Tutoring System. *Intelligent tutoring systems: At the crossroads of artificial intelligence and education*, 83–106.

Crapo A and Moitra A (2013) Toward a unified English-like representation of semantic models, data, and graph patterns for subject matter experts. *International Journal of Semantic Computing* **07**, 215–236.

Dzikovska M, Steinhauser N, Farrow E, Moore J and Campbell G (2014) BEETLE II: deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *International Journal of Artificial Intelligence in Education* **24**, 284–332.

Ferreira A and Atkinson J (2009) Designing a feedback component of an intelligent tutoring system for foreign language. *Knowledge-Based Systems* **22**, 496–501.

Gutierrez F and Atkinson J (2011) Adaptive feedback selection for intelligent tutoring systems. *Expert Systems with Applications* **38**, 6146–6152.

Jeon SM, Lee JH, Hahm GJ and Suh HW (2016) Automatic CAD model retrieval based on design documents using semantic processing and rule processing. *Computers in Industry* **77**, 29–47.

Kang J and Lee JK (2005) Rule identification from Web pages by the XRML approach. *Decision Support Systems* **41**, 205–227.

Kang S, Patil L, Rangarajan A, Moitra A, Jia T, Robinson D and Dutta D (2015) Extraction of manufacturing rules from unstructured text using a semantic framework. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V01BT02A033–V01BT02A033). American Society of Mechanical Engineers.

Keuning H, Jeuring J and Heeren B (2016) Towards a systematic review of automated feedback generation for programming exercises – extended version. *Technical Report Series*, (UU-CS-2016-001).

Lane HC and VanLehn K (2005) Teaching the tacit knowledge of programming to noviceswith natural language tutoring. *Computer Science Education* **15**, 183–201.

Le NT and Pinkwart N (2011) INCOM: A web-based homework coaching system for logic programming. In *Conference on Cognition and Exploratory Learning in Digital Age*. pp. 43–50.

Li Z and Ramani K (2007) Ontology-based design information extraction and retrieval. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **21**, 137–154.

Li Z, Liu M, Anderson DC and Ramani K (2005) Semantics-based design knowledge annotation and retrieval. In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 799–808). American Society of Mechanical Engineers.

Li Z, Raskin V and Ramani K (2007) A methodology of engineering ontology development for information retrieval. In *Proceedings of the 16th International Conference on Engineering Design (ICED'07)*.

Li Z, Yang MC and Ramani K (2009) A methodology for engineering ontology acquisition and validation. *AI EDAM* **23**, 37–51.

MacNish C (2002) Machine learning and visualisation techniques for inferring logical errors in student code submissions. In *ICALT-2002: Proc. 2nd IEEE Int. Conf. on Advanced Learning Technologies* (pp. 317–321).

Mitrovic A, Suraweera P, Martin B and Weerasinghe A (2004) DB-suite: experiences with three intelligent, web-based database tutors. *Journal of Interactive Learning Research* **15**, 409.

Nagata N (2002) Banzai: Computer assisted sentence production practice with intelligent feedback. *Computer assisted system for teaching and learning Japanese, 2002*.

Ohlsson S (1994) Constraint-based student modeling. In Greer JE and McCalla GI (eds), *Student Modelling: The Key to Individualized Knowledge-Based Instruction*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 167–189.

OWL: Web Ontology Language (2011) *OWL: Web Ontology Language*. Berlin/Heidelberg: Springer-Verlag.

Pinquie R, Veron P, Segonds F and Croue N (2015) Natural Language Processing of Requirements for Model-Based Product Design with ENOVIA/CATIA V6. In *IFIP International Conference on Product Lifecycle Management* (pp. 205–215). Springer, Cham.

Rangarajan A, Radhakrishnan P, Moitra A, Crapo A and Robinson D (2013) Manufacturability analysis and design feedback system developed using semantic framework. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V004T05A001–V004T05A001). American Society of Mechanical Engineers.

RDF 1.1 Concepts and Abstract Syntax (2014) *RDF 1.1 Concepts and Abstract Syntax*. W3C—World Wide Web Consortium. Available at https://www.w3.org/TR/rdf11-concepts/.

RDF Schema 1.1. (2014) *RDF Schema 1.1*. W3C—World Wide Web Consortium. Available at https://www.w3.org/TR/rdf-schema/.

Riel A and Boonyasopon P (2009) A knowledge mining approach to document classification. *The Asian International Journal of Science and Technology in Production and Manufacturing* **2**, 1–10.

Shotorbani PY, Ameri F, Kulvatunyou B and Ivezic N (2016) A hybrid method for manufacturing text mining based on document clustering and topic modeling techniques. In *IFIP International Conference on Advances in Production Management Systems* (pp. 777–786). Springer, Cham.

The Natural Language Processing for JVM languages (NLP4J). Retrieved July 4, 2017, Available at https://emorynlp.github.io/nlp4j/.

Ur-Rahman N and Harding JA (2012) Textual data mining for industrial knowledge management and text classification: a business oriented approach. *Expert Systems with Applications* **39**, 4729–4739.

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press

Wang G, Tian X, Geng J, Evans R and Che S (2016) Extraction of principle knowledge from process patents for manufacturing process innovation. *Procedia CIRP* **56**, 193–198.

Yang H, De Roeck A, Gervasi V, Willis A and Nuseibeh B (2011) Analysing anaphoric ambiguity in natural language requirements. *Requirements engineering* **16**(3), 163–189.

Yang MC, Wood III WH and Cutkosky MR (2005) Design information retrieval: a thesauri-based approach for reuse of informal design information. *Engineering with Computers* **21**, 177–192.

Yu L, Wang S and Lai KK (2005) A rough-set-refined text mining approach for crude oil market tendency forecasting. *International Journal of Knowledge and Systems Sciences* **2**, 33–46.

Dr. SungKu Kang obtained his B.S. in Mechanical and Aerospace Engineering from Seoul National University in 2012, and his Ph.D. in Mechanical Engineering from the University of Illinois at Urbana-Champaign in 2017. During his Ph.D., He has collaborated with GE Global Research as a Graduate Research Assistant. He has been involved in the research projects including formal manufacturing rule extraction from unstructured English text utilizing Natural Language Processing (NLP) and semantic technology (i.e., ontology). The value of his researches lies in the development of formal manufacturing knowledge that a computer can "understand" to perform complex manufacturing-related tasks, such as manufacturability analysis. Currently, he is doing his research as a Postdoctoral Research Associate at Virginia Polytechnic Institute and State University, and his research interest is extended to the application of Data-driven methods and Data-fusion model along with NLP and Semantic technology to realize design automation and innovate manufacturing process.

Dr. Lalit Patil received this Ph.D. from the University of Michigan at Ann Arbor. He previously worked at the University of Illinois at Urbana-Champaign and the University of Michigan, Ann Arbor. Currently, he is working as a senior solution architect at Ford Motor Company. His research interests are in the area of innovation, and in the management of information and knowledge to support processes, methods, and tools for product creation.

Dr. Arvind Rangarajan is a technical leader – additive/automation for GE Renewable Energy. He is currently working on advanced manufacturing technologies including additive design tools, model-based manufacturing, and robotics. He has a doctorate in mechanical engineering from University of California at Berkeley.

Dr. Abha Moitra is a Principal Scientist at GE Global Research in Niskayuna, NY. Her research interests are in Semantic Modeling and Reasoning, Knowledge Management, Open Source Intelligence, Social Network Analysis, Natural Language Processing and Optimization. She has over 25 years of experience working for GE Research during which she had led the development of numerous systems that are currently in use at several current and former GE businesses. Most recently, her work has focused on semantic models and reasoning to capture domain knowledge and deployment in decision support systems. She has developed semantic applications in a broad spectrum of domains, including cyber resiliency, requirements capture and analysis, manufacturability, electric grid semantic model built on NIST's Common Information Model, system for capturing and analyzing data provenance and trust in a multi-level secure environment, drug-drug interactions. Dr. Moitra is active in the Semantic community.

Dr. Tao Jia works in the Advanced Manufacturing Engineering Technology Center at General Electric Healthcare in Waukesha, Wisconsin. Her responsibilities include leading digital manufacturing effort to reduce manufacturing cost and applying data analysis to improve manufacturing process. She received a master's degree in computer science in 2012 and a PhD in mechanical engineering in 2012 from Michigan State University. She also holds a bachelor's degree in electrical engineering from Shanghai Jiao Tong University. Her research interests are computer-aided design and manufacturing (CAD/CAM) related to direct digital manufacturing and additive manufacturing. Jia has done seminal work on enabling design for manufacturing via integration of digital thread, driving cost reduction and time-to-market. She is also working on promoting model-based definition to manufacturing process, enabling integration of product and manufacturing information in to CAD models. She is a recipient of 2017 SME Outstanding Young Manufacturing Engineer Award and she is actively involved in the GE women's network to promote women's presence in manufacturing field and to foster girls' interest in STEM.

Dr. Dean M. Robinson is a Principal Engineer in Additive Manufacturing Technologies at GE Research in Niskayuna, NY, where he leads projects in geometry modeling, physics-based process simulation, and build preparation software for additive manufacturing. From 2000 to 2017, Dr. Robinson was lab manager for model-based manufacturing at GE Research. His lab developed technologies and software for 3-D model-based manufacturing, variation management, and geometry-process integration in precision manufacturing, inspection & repair, working with several GE businesses. Key focus areas included design for producibility, adaptive & robotic machining, and distortion correction for additive manufacturing. From 2005 to 2010, Dr. Robinson's lab launched several projects in additive manufacturing, acquiring or building machines to manufacture parts in materials including ceramics, polymers and casting wax. He has worked at GE Research since 1983, except for a 5-year leave to pursue full-time graduate study. Dean earned a B.S. in Mechanical Engineering from Princeton University, an M.Eng. in Mechanical Engineering from Rensselaer Polytechnic Institute, and a PhD in Mechanical Engineering from Cornell University.

Dr. Debasish Dutta received his Ph.D. degree from Purdue University. He is a Distinguished Professor of Engineering and the former Chancellor at Rutgers University–New Brunswick. Previously, he was with Purdue University, University of Illinois at Urbana-Champaign, and the University of Michigan, Ann Arbor. He is an Elected Fellow of the American Association for the Advancement of Science and American Society of Mechanical Engineers. His research interests are in the area of innovation and lifelong learning, and product data and its central role when managing informational needs throughout the product life cycle.

https://doi.org/10.1017/S0890060419000027 Published online by Cambridge University Press