

Open the Pod Bay Door: Using Ontology to Understand Instructions¹

Yixin SUN^a, Michael GRÜNINGER^a,

^a*Department of Mechanical and Industrial Engineering, University of Toronto, Ontario,
Canada M5S 3G8*

Abstract. There have been a great deal of developments and implementations for conversational robots like Amazon Alexa, and Google Home. Yet, little research has been done for robots to understand a more rigorously structured, instruction-wise natural language. In this paper, we present an ontology approach to convert instructions from natural language to logical formulas, Process Specification Language (PSL) in this case. Verbs, therefore, are treated as indication of actions or processes and are decomposed semantically to understand meanings. As a method of evaluating this approach, verbs, originally in PSL, are worked towards natural language. In this paper, we present how to take natural language instructions and map them to appropriate cutting classes; and moreover, we present the capability of this ontology-centered approach to go in a reverse direction (from PSL to natural language).

Keywords. process ontology, semantic parsing, instructions, cutting process

1. Introduction

In the movie, *2001: A Space Odyssey*, the ship's computer Hal would not follow Dave's spoken instructions, and did not open the pod door. When someone utters "open the pod bay door", indeed, only the end result is specified, lacking all of the detailed steps in between. This door example best mimics human-to-human instruction, as opposed to the way that programmers will hard-code an autonomous robot, stressing even the robot's exact angle with the door knob. This paper focuses on a higher level of abstraction and thus studies how a robot can understand instructions uttered in natural language. The form of matters we have based our research on is verbal instructions, which is inspired by the notion of Physical Turing Test.

The Physical Turing Test extends the Turing Test by specifying questions that require the integration of perception, reasoning, and action. In particular, Ortiz et al [1] propose two tracks for the Physical Turing Test. The Construction Track focuses on building predefined structures (such as a tent or modular furniture) given a combination of verbal instructions and images. The research presented in this paper is motivated by the following long-term vision related to the Construction Track:

¹Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Given a set of verbal instructions, together with a sequence of annotated images, answer questions about the activities that can possibly occur during performance of the instructions and the various objects that participate in these activity occurrences.

In particular, we write out instructions not in a step-by-step manner, but in the most usual way for a worker or a practitioner to follow. By mapping natural language to the Process Specification Language (PSL) Ontology [2], we approach the problem by matching action verbs in a sentence to a particular process and then mapping this process to a first-order logic formula. Understanding verbs becomes our primary goal moving forward. Furthermore, in order to validate whether the process generated from natural language is correct or not, we can invert this approach and remap the logical formula - that is the PSL process description - to natural language. This is a proof-of-concept paper that proposes how semantic parsing can work from an ontology-focused perspective and verifies its feasibility.

2. Literature Review

The Alexa Meaning Representation Language (AMRL) [3] introduced a graphical hierarchy of the main Amazon Alexa ontology with nodes and arcs. Classes (each represented by a node) are connected with a property (an arc) in between, while children classes are connected to parent classes. In this way, Amazon Alexa can understand a more complex and cross-domain conversation. In this study, Amazon Alexa had better performance in solving ambiguities when mapped to AMRL, rather than spoken language understanding. This study mainly sought solutions for flexibility and ambiguity in which its approach could be useful to our study. Nonetheless, our focus differs from Amazon Alexa insofar as we are not interested in daily conversational languages; instead, we are seeking to understand a more formally written language which can be best represented by the category of instructions.

According to Levin's definition of English Verb Classes [4], verbs can be divided into four categories: hit verb, touch verb, cut verb, and break verb. These classes differ by whether a change of state occurs, whether a contact with another object is made, and whether it in nature is a causative/inchoative motion, respectively. Under each category lies different subclasses, categorized by whether or not the subclass passes the designated alteration tests. Based on the nature of the tests, verbs belong to the same classes and subclasses do not necessarily imply they are synonyms. Classifying verbs in this way can only provide guidance to syntactic uses. VerbNet [5] further extends Levin's idea to a coding library that can be imported into Java/Python/C++. However, this package is not comprehensive enough to be incorporated into our study as each subclass has a limited number of sample verbs.

Nonetheless, Levin has pointed out a possible approach for instruction mapping: verb synonyms under the same subclasses can be used in the same sentence and convey the same meaning. In this way, The Cutting Process Ontology [6] proposed nine different ways of cutting a two-dimensional piece of sheet metal. The paper introduced three new topological components: points, edges and surfaces. An edge is a subset of at least two points, while a surface is a subset of at least two edges. Furthermore, other definitions can be built on top of the existing components. For instance, a hole is a hollow surface

within a solid one. The PSL Ontology was used to axiomatize the cutting processes in terms of whether any edges, holes and/or surfaces were created/destroyed (see Figure 1).

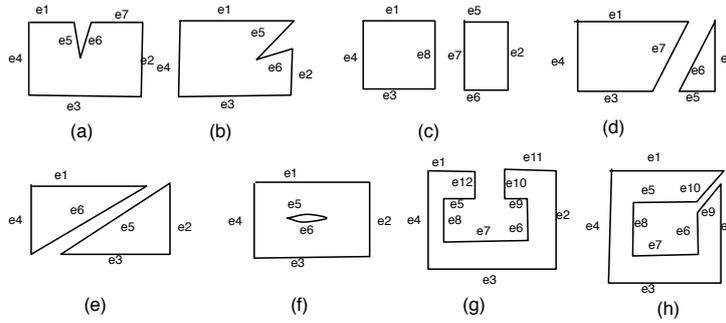


Figure 1. Classes of cutting processes.

Through mapping natural language expressions to the Cutting Process Ontology, synonyms of “cut” such as “chip”, “split”, “snip” all belong to the same subclass, indicating that these synonyms can be represented by one or more of the cutting classes identified in this paper. This indicates that synonyms can share some of the existing ontology axioms. When it comes to translating instructions into PSL, this finding can be useful as fewer new terms need to be created.

Furthermore, physical cutting activities, in nature, are not limited to two-dimensional shapes. Some of the instructions may require cutting action to be done on three-dimensional shapes when it comes to tree trunks or food recipes. There also lie some intangible objects like movies, words and genes that when the cutting action were to be performed on them, they would not change shape; instead they would change sequence-wise. To better describe all cutting processes and instructions, extending the cutting ontology to include three-dimensional shapes can be considered for future work.

3. Approach

We focus on developing an ontology-specific semantic parser, in which treats the action verb as an utterance of processes and convert that into a first-order logic process description. For evaluation and verification purposes, our approach can be proved by going backwards in this pipeline. We can test the natural language instructions deduced from the axioms against what people understand from natural language. Whether the two utterances yield the same meaning can be treated as an evaluation criteria. The verb, “cut” is taken as an example to demonstrate this process flow. See Figure 2 for details.

3.1. Semantic Analysis in the context of PSL

A PSL description is a logical formula that can specify an occurrence of activity in terms of time-points while capturing changes in properties throughout the activity. We consider the **verb** in a natural language sentence to be the representation of the **activity** that causes objects to change state. In other words, only when something is **done** will the **subjects**

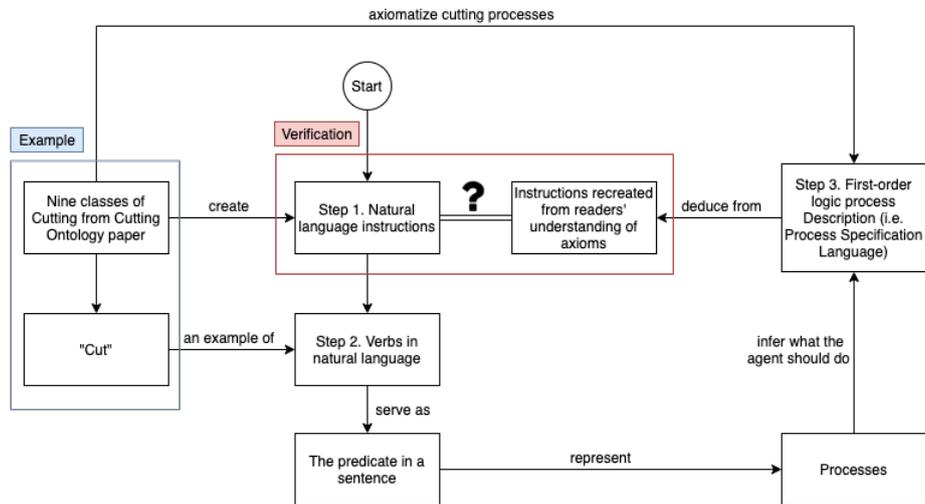


Figure 2. Process Flow Diagram of Semantic Parsing.

be affected. Therefore, the first thing to do is to locate where a verb is and of which two words it falls in between; this can determine the primitive activity and the agent and/or operator. A syntactic parsing tool, SpaCy [7], was applied to identify part-of-speech tags.

3.2. Cutting Ontology

After natural language processing by SpaCy, we examine verbs specifically to interpret their intended semantics. Motivated by the Cutting Process Ontology, topological instructions were created based on existing PSL ontology. Since cutting is the verb in all of the written instructions, by translating them into PSL, we hoped to find some patterns that could apply universally in terms of converting instructions into logical formulas. Therefore, the primary findings that will be presented in this paper centers around the physical action verb, "cut". Figure 1 illustrates the nine different classes of cutting activities and the corresponding changes to the shape of the object.

3.3. Cutting Instructions

In natural language, cutting is an action that does not require much description. It all comes in naturally when a human were to cut something. This is evident in the data collected from WikiHow.com² in which physical cutting activities do not stand alone as an individual instruction of multiple steps to follow. Instead, the action of cutting is mentioned in 193 different activities ranging from "cutting onions", "cutting bread" to paper cutting. Yet, ambiguity exists when the instructions given were "Cut a piece of sheet metal". Questions like "in what shape should the end result be" arise. Therefore, to better examine the backward loop of converting natural language into PSL, we created

²Instruction data was scraped off from the website *WikiHow.com* as it was considered to be a go-to place for real-life problems, ranging from math questions to furniture assembly and maintenance. In total, 922 rows of data were scraped from the featured article page on May 6, 2020.

instructions in respect to the nine classes of cutting from a topological perspective. That is, only relationships between edges, surfaces and holes were explained. Table 1 shows the finalized written instructions with respect to the nine classes of the Cutting Process Ontology [6].

Table 1. Cutting Instructions from a topological perspective. The letters in the first column refer to Figure 1

Illustration No.	Written Instructions
a	On one edge of the rectangle, from any point excluding the corners, cut so that three additional edges can be created, while creating no extra surfaces or holes.
b	From one corner of the rectangle, make a cut so that two additional edges are created, while creating no extra surfaces or holes.
c	Make a cut starting from any point (excluding the corners) of the rectangle and ending at any point (excluding the corners) on the opposite edge. The end result should be two separate surfaces that make up four additional edges. No extra holes should be created.
d	Starting from any corner of the rectangle and ending at any point excluding the corners on the opposite edge, the cut should create two separate surfaces from the rectangle with three additional edges. No extra holes should be created.
e	From any corner of the rectangle, make a cut that ends on the opposing corner. Two separate surfaces and Two additional edges should be created. No extra holes should be created.
f	From any point inside the rectangle, make a cut that is a closed-curve. The cut should not reach any edges of the rectangle. Two additional edges should be created. An additional edge should be created. No extra surfaces should be present.
g	From any point inside the rectangle, make four cuts that create four new edges that do not intersect with any of the existing edges. A hole should be created as a result. No extra surfaces should be present.
h	Continue from h), destroy the existing hole. Make two cuts starting from the width of the outer rectangle, ending at two points (excluding corners) of the closest inner edge. Four additional edges should be created. No extra surfaces should be created.
i	Continue from h),destroy the existing hole. Make a cut (Cut E) starting from any corner (Corner A) of the outer rectangle, ending at a corner (Corner B) of the inner shape. This cut should not cut through any part of the rectangle that does not have any material. Make a second cut (Cut F) starting from any point on the outer edge in which Corner A intersects, ending at any point on the inner edge in which Corner B intersects. This cut should not cut through any part of the rectangle that does not have any material. Two additional edges should be created. No extra surfaces should be created.

4. Shape Cutting Ontology

Since PSL specifies a change of state, locations or directions, the starting position mentioned in the written instruction were neglected. Only **starting state** and **ending state** were taken into account. Following the open-the-door example mentioned in section 1, human instructions are usually given in a higher level of abstraction. Correspondingly, the classes of the Cutting Process Ontology were applied to instructions shown in Table 1, and results can be found in Table 2.

Table 2. Axiomatization of cutting processes. The letters in the first column refer to Figure 1 and the axioms in the second column use classes from the Cutting Process Ontology.

Cutting Process	Axiomatization
a	$preserve_surface(a) \wedge preserve_hole(a) \wedge change_one_meet(a) \wedge create_three_edge(a)$
b	$preserve_surface(a) \wedge preserve_hole(a) \wedge change_one_meet(a) \wedge create_two_edge(a)$
c	$create_surface(a) \wedge preserve_hole(a) \wedge change_one_meet(a) \wedge create_four_edge(a)$
d	$create_surface(a) \wedge preserve_hole(a) \wedge change_one_meet(a) \wedge create_three_edge(a)$
e	$create_surface(a) \wedge preserve_hole(a) \wedge change_one_meet(a) \wedge create_two_edge(a)$
f	$preserve_surface(a) \wedge destroy_hole(a) \wedge preserve_meet(a) \wedge create_two_edge(a)$
g	$preserve_surface(a) \wedge destroy_hole(a) \wedge change_one_meet(a) \wedge create_four_edge(a)$
h	$preserve_surface(a) \wedge destroy_hole(a) \wedge change_two_meet(a) \wedge create_four_edge(a)$
i	$preserve_surface(a) \wedge destroy_hole(a) \wedge change_one_meet(a) \wedge create_two_edge(a)$

4.1. Discussion

The instructions that we were able to recreate from the given illustrations have only included the starting and ending state of the shape. As a result, only topological relationships were involved, describing whether an edge, a surface, and/or a hole were to be preserved or destroyed throughout the process. Yet, geometric relationships could also be incorporated, making up a more detailed and thus more robotic-oriented instruction. For instance, a geometric way of describing illustration (a) would be: On one width of the rectangle, from any point excluding the corners, make a cut towards the centre of the rectangle. The cut should not be parallel to the height of the rectangle. Stop before the cut reaches the opposing width. From the end point of the cut, make another cut that ends on any point of the original width excluding corners. This would be a detailed version of **how** to make a cut, instead of only stressing the starting and ending conditions. Geometric concepts are largely introduced here: “towards the centre”, “parallel to height”, and etc. If this were to be the simplest version of cutting instructions, then the cutting ontology would need an upgrade to include geometric definitions.

5. Conclusion

This paper presents our initial efforts towards mapping natural language instructions to process descriptions using the PSL Ontology. In particular, cutting processes are used as a demonstration of our approach. Meanwhile, as we explore mapping procedures, we have found that going in the other direction - from PSL process descriptions to natural language - can help us understand how instructions should be written and interpreted. For example, in a day-to-day work routine, human beings communicate instructions in a sense that specify end results, yet inevitably neglecting how to make a cut. “*How is cutting a sheet metal different from cutting a tree trunk?*” is one possible question that can arise in a physical Turing Test.

Instructions, as a form of written natural language, tend to be more straight-forward and less metaphorical. As for future studies, conversational natural language can be studied that incorporate more rhetorical devices. For cutting instructions specifically, its ontology could be extended in such a way that three-dimensional and intangible objects will be included to accommodate a wider range of cutting instructions.

References

- [1] Ortiz, Jr, C. L. (2016). Why We Need a Physically Embodied Turing Test and What It Might Look Like. *AI Magazine*, 37(1), 55-62.
- [2] Gruninger, M. (2003) Ontology of the Process Specification Language, pp. 599-618, *Handbook of Ontologies and Information Systems*, S. Staab (ed.). Springer-Verlag
- [3] Kollar, T., Berry, D., Stuart, L., Owczarzak, K., Chung, T., Mathias, L., Kayser, M., Snow, B., Matsoukas, S. (2018) The Alexa Meaning Representation Language. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)
- [4] Levin, B. (1993) English Verb Classes and Alternations: A Preliminary Investigation. University of Chicago Press.
- [5] Brown SW, Bonn J, Gung J, Zaenen A, Pustejovsky J, Palmer M. (2019) VerbNet Representations: Subevent Semantics for Transfer Verbs. Proceedings of the First International Workshop on Designing Meaning Representations.
- [6] Gruninger, M. and Delaval, A. (2009) A First-Order Cutting Process Ontology for Sheet Metal Parts, *Fifth Conference on Formal Ontology Meets Industry*, Vicenza, Italy.
- [7] Choi, J., Tetreault, J., Stent, J. (2015) It Depends: Dependency Parser Comparison Using a Web-based Extraction Tool. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 387–396. Beijing, China, July 26-31, 2015.