Information Extraction on High-School Level Chemistry Labs

Technical Report TR2010-929

Daniel Galron
Computer Science Department
Courant Institute of Mathematical Sciences
New York University
galron@cs.nyu.edu

This work is supported by NSF grant IIS-0534809

Abstract

In this report we present a feasibility study on automatically interpreting instructions found in a set of high school chemistry labs, and discuss the role of deep domain knowledge in the interpretation. We define the task of sentence-level interpretation as the extraction of symbolic representations of the sentence semantics. In the broader scope, the sentence-level semantics of a particular sentence will be resolved with semantics from other sentences in the lab along with domain knowledge to disambiguate and reason about a physical system. The task of general automatic sentence-level interpretation is a difficult one. The general problem is not very well defined in the natural language processing research community, and few researchers have studied the problem. The common practice is to decompose the problem into subtasks, such as resolving coreferences of noun phrases, labeling the semantic roles of arguments to predicates, and identifying word categories. We describe a pipeline combining the subtasks described, along with parsing, to create a system capable of extracting sentence-level semantics. All the systems used for the subtask are found off-the-shelf, and we should stress that such a system will be highly-error prone for reasons we discuss below. Finally, we do a close study of the chemistry lab corpus, and analyze each instruction to determine the feasibility of its automatic interpretation and the role of deep domain knowledge in its disambiguation and understanding.

Introduction

The goal of the project is to interpret information and instructions found in high school level chemistry labs and construct a framework to allow for automated reasoning about chemistry problems.

The problem of general text interpretation is not well defined in the natural language processing research community. The task is generally broken down into trying to learn certain types of textual information – for instance, identifying and labeling named entities (persons, organizations, countries, and so on), analyzing the syntactic structure of sentences (i.e. syntactic parsing), discovering the semantic relationships between words in a sentence (i.e. dependency parsing and semantic role labeling), and so on. For the purposes of text interpretation for automated reasoning, we need to construct symbolic representations of the semantics of instructions and constraints found in the labs. The general problem of automatically constructing complete semantic representations of sentences is not very well studied (although there are a few exceptions.) To my knowledge, there is no complete off-the-shelf system that does so, and there are standard evaluation measures to determine how accurate the semantic representations are, due to the complexity of the problem. To automatically construct semantic representations of the test, we would need to utilize systems that automatically solve some of the subtasks. In the first part of this report, I discuss some of the research literature on the subtasks, and discuss the feasibility of utilizing them to construct the semantic representations. In the second part, I will perform a close study of the chemistry lab instructions, discuss the feasibility of a hypothetical system constructing a semantic representation of the text, and discuss the role of deep domain knowledge in interpreting each instruction.

Overview

Broadly speaking, labs have a standard format. The first section is a motivation of the lab and a discussion of the chemical principles at play. Relevant technical terms are usually defined, as well as formulae detailing the chemical reactions occurring the lab. They may review some safety procedures, but the primary purpose is to motivate the lab and explain the chemical principles behind it. Following the exposition is the lab procedure. This is typically an ordered list of steps to take in the lab. Each step consists of imperative instructions, such as “Heat the test tube,” or “Empty the beaker but do not dry it.” A list item may contain several imperative statements, so we cannot assume a one-to-one correspondence between list items and steps in the lab. The last sections of the lab tend to be instructions for data analysis and questions on the labs.

The goal of this work is to be able to achieve machine understanding and possible execution of the instructions given in the lab. In the following section I will describe the instructions in three examples of a lab studying the decomposition of potassium chlorate. I will break down the instructions and discuss the deep domain knowledge needed to disambiguate and understand them, as well as the domain knowledge that could possibly be extracted from them.

Techniques necessary to interpret the labs

Using deep domain knowledge may very well help in interpreting and understanding the text, namely in disambiguating procedures and concepts. When interpreting the text, there are two levels we must consider: sentence-level interpretation, and document-level interpretation. Sentence-level interpretation involves disambiguating the sentences and constructing a symbolic representation of their semantics. In effect, on this level, we wish to interpret each sentence independent of the other sentences. From sentence-level interpretations, it would be possible to extract deep domain knowledge that may help us in interpreting on the document level. Once we do so, we wish to interpret the document as a whole, that is, we would like to infer the semantics of sentence sequences. In the context of interpreting high-school level Chemistry labs, this would involve reasoning about the semantics of the instructions in the lab, for instance, reasoning about the constraints applied to one instruction but specified by another. Sentence-level interpretation can be done by utilizing several techniques from natural language processing to construct symbolic representations of each sentence that an inference engine could then work with. To interpret on the sentence level, for each instruction, constraint, or statement in the text we would need to identify the predicates, identify the arguments to the predicates and their type, resolve the instances of the entities the arguments map to, and establish the causal or temporal links between the statement and the other statements and instructions in the lab. This would require several steps, including: identifying noun phrase entities as being chemicals, equipment, or physical/chemical properties; interpreting instructions or other verb phrases as relationships between entities or describing the states of the entities; and resolving references to instances of the entities. From the sentence-level interpretations, we can extract some domain knowledge, and use that domain knowledge, perhaps with domain knowledge from other sources, to reason about the implications of each instruction. In the following sections, we describe methods for solving each of the sub-tasks above associated with sentence-level interpretation, as well as potential challenges and pitfalls.

Identifying Entities

An important component of extracting the semantic representation is to identify and classify the equipment, chemicals, and concepts that are being manipulated. Since the goal of sentence-level interpretation is to construct a symbolic representation of a sentence's semantics, the types of the entities being manipulated are crucial. This task is akin to doing *named entity recognition (NER)*. In the NER task, the goal is to label named entities in a text with their type. These types typically include person names, organization names, dates, times, numbers, percentages, and locations. In bioinformatics, the types may include protein, DNA, RNA, cell line, and so forth(Nadeau & Sekine, 2007).

The dominant method used for the NER task is to apply a machine learning algorithm to learn the named entity types from a large training corpus, and then applying the learned model to classify the named entities in previously unseen texts. There are three flavors of ML algorithms applied to the problem. In *supervised learning methods*, model parameters are trained on a large annotated corpus to disambiguate between categories for named entities based on the correlated features. Supervised methods tend to be the baselines for most of the work done in NER today. For many domains, they achieve state-of-the-art performance. In *semi-supervised learning methods*, a supervised learning algorithm or a set of manually constructed rules is used to bootstrap a model from an annotated corpus, and uses that model to accumulate information about the context in which the named entities are found. Then, the algorithm searches for similar contexts in unannotated corpora, and uses the contexts found to identify the named entities and their types. Finally, in *unsupervised learning methods*, *clustering* is used to find named entities and categorize them according to the distributional similarity of their contexts. The contexts can be identified by using hyponyms/hypernyms (Evans, 2003), by using co-occurrence statistics along with WordNet Named Entity categories (Alfonseca & Manandhar, 2002), and by using Pointwise Mutual Information and Information Retrieval (Etzioni, et al., 1005).

For this task, it would be useful to define three types: equipment (e.g. test tube, beaker, Bunsen burner), chemical (e.g. Potassium Chlorate, Manganese Oxide), and chemical property (e.g. mass, volume, pressure). There has been little work done in NER for chemical identification. (Narayanaswamy, Ravikumar, & Vijay-Shanker, 2003) apply a set of rules and heuristics to augment an NER system for protein identification with chemical identification. They manually created a set of rules relying on lexical and contextual information to identify chemical names. In their approach, they could not use supervised learning due to a lack of high-quality annotated data. For the task of text interpretation of high-school level chemistry lab, using machine learning method would probably be unfeasible. Supervised learning methods would require us to have annotated data, and semi-supervised and unsupervised learning method would require us to have very large (unannotated) corpora in order to gather the requisite context statistics. There is an advantage, however, in our domain – there is little ambiguity when it comes to names. In a document containing person named entities and organization named entities, if the word "Ford" appeared, it may be difficult to discern whether it refers to the automotive company or a person. In the chemistry lab domain, there is little ambiguity between equipment, chemicals, and properties. In order to categorize the entities with a high degree of accuracy, we use dictionaries in conjunction with rules for identification. The equipment and properties categories consist of more-or-less closed vocabularies. It should be possible to compile a list of equipment and a list of properties with minimal difficulty. To identify chemicals, we could use one of several online chemical databases, such as PubChem or ChemSpider. Alternatively, we could devise a set of rules to help us identify chemicals. For instance, a regular expression query such as ([<CH>]+[1-9]*) (where <CH> is a list of chemical symbols) would find all chemical formulas in the text. Similarly, we could search for word sequences that consist of chemical names (e.g. "Potassium Chlorate", or "Manganese Oxide"), or for words that have certain morphemes (e.g. "-oxide", "-yl", "-ate"). Due to the regularities in chemical naming conventions, these sorts of heuristics would likely identify the chemical names correctly with high accuracy.

Constructing predicates and arguments

Once we identify the entities and concepts being manipulated, we need to construct symbolic representations of the instructions and informational statements in the text. This requires a semantic analysis of each statement, identifying the predicates (i.e. the action to be performed), the arguments the predicate takes (i.e. what are the entities in the interaction), and identifying the instances of the entities being manipulated. To make this more concrete, lets take as an example the third instruction in step 4 of the lab "The Decomposition of Potassium Chlorate." The instruction says:

> *Remove the stopper from the test tube.*

The semantic analysis of the instruction would be something like:

> [$_{pred}$ Remove] [$_{arg0\text{-}theme\text{-}equip}$ the stopper] from [$_{arg1\text{-}patient\text{-}equip}$ the test tube]

meaning that the stopper is the first argument and is a piece of equipment, and that the test tube is the second argument and is also a piece of equipment. This would have a symbolic representation similar to:

> REMOVE(STOPPER$_0$, TEST_TUBE$_1$)

Where the predicate takes arguments:

REMOVE(*arg0-equip, arg1-equip*)

Two techniques were utilized in this semantic analysis; first, we identified the predicate and the arguments the predicate takes (as well as the type of the arguments); and second, we resolved the arguments to instances of particular entities. If the stopper and the test tube had been used in an earlier instruction, then the coreference resolution system would map those arguments to the previously identified instances.

To extract the symbolic representations of sentence semantics, we need to construct a syntactic or dependency parse of the sentence that captures the relationship between the lexical items in the sentence. This step is a prerequisite of labeling the predicates and arguments in the sentence. In the next section I briefly discuss some successful approaches to parsing (giving references to the relevant literature) and introduce some off-the-shelf systems we could use.

Parsing

The semantic "meaning" of a sentence is a function of the sentence structure – once we know the structure of a sentence, we have a model of how the words in the sentence relate to each other, and can compose the sentence-level semantics from the structure on the word-level semantics. There are two types of parses we can try to infer for sentences – syntactic parses and dependency parses. Syntactic parses bracket the grammatical constituents in the sentence into phrases, and describe the structure of the sentence according to these phrases. Dependency parses capture the relationships between words based on their meaning. In dependency parses, the structure of the sentence is captured by detailing the relationship between *head words* and their dependents (e.g. verbs and their subject or object arguments). A dependent in one head-dependent relation may be the head of another head-dependent relation.

The most successful parsers are stochastic parses that find the highest-scoring parse of a sentence according to a model. The parameters of the model are set by a machine learning algorithm to optimize an objective function that is typically a function of the number of correct bracketings (in the case of syntactic parses) or dependencies (in the case of dependency parsers). Some notable current state-of-the-art syntactic parsing methods include (Taskar, Klein, Collins, Koller, & Manning, 2004), (Bikel, 2004), (Collins & Koo, 2005), (Turian, Wellington, & Melamed, 2006) and (Huang, 2008). Less work has been done on dependency parsing, but some recent work, such as (McDonald, Pereira, Ribarov, & Hajic, 2005) and (Nivre, 2007).

Given that parsing is a very well studied problem, many off-the-shelf parsers are available for use. Two high-performing ones are the Bikel parser and the Stanford parser. The Bikel parser (Bikel, 2004) (http://www.cis.upenn.edu/~dbikel/software.html) is a high-performing syntactic parser based on the work of Michael Collins (Collins 1996, 1997, 1999). It is trained on the Wall Street Journal sections of the Penn Treebank, and outputs reasonably high quality parses on similar corpora (i.e. newswire). The Stanford parser (Klein & Manning, 2003) (http://nlp.stanford.edu/software/lex-parser.shtml) provides both syntactic parses *and* dependency relations between head words and depenents. Its English parser too is trained on the Penn Treebank. Whether it will perform sufficiently well in the chemistry lab domain is an empirical question that can only be answered experimentally.

Semantic role labeling

The goal of semantic role labeling is to identify predicates (typically verbs, or another lexical item associated with an action) and their arguments, and to label those arguments with the *role*. Examples of the role include the *agent* (the one performing the action), the *patient* (the object of the action experiencing a change of state), the *location*, and so on. This is done by constructing a syntactic analysis of the sentence (usually represented as a parse tree), and labeling the nodes that immediately dominate the predicate and arguments. Conceptually, this is a multi-label classification problem in much the same way as named entity recognition, and so machine learning methods are typically applied to the problem. Although we're able to get away from using ML techniques for named entity recognition applied to the chemistry lab corpus, we will not be able to simply write a series of rules to label the predicates and arguments. Their vocabulary is not closed, and the dependencies between the labels of the predicates and arguments are complex. We would need to rely on the syntactic structure of the instructions in order to disambiguate between labels, as well as how the other arguments to a particular predicate were labeled. In this section, we describe some of the techniques applied to the task, and how we could leverage them to interpret the chemistry labs.

Most of the work done in semantic role labeling involves training models to choose the labels. Historically, the model is factored into two steps; the first selects which of the nodes in the parse tree of the sentence dominate the predicate and which ones dominate the arguments, and the second labels each argument with its semantic role. These systems are typically trained in a supervised manner on the PropBank (Palmer, Gildea, & Kingsbury, 2005), FrameNet (Fillmore, Johnson, & Petruck, 2002), and VerbNet(Kipper-Schuler, 2005) corpora. The challenge of building such systems involves training them to accurately label the nodes, and to be robust in the face of parser error. The most common approach involves learning how to label the nodes in a syntactic analysis of a sentence that models the dependencies between the arguments of a predicate. The most successful (i.e. best-performing) systems attempt to learn argument labels jointly rather than individually, since there are inherent dependencies between the different semantic roles of the arguments. (Gildea & Jurafsky, 2002) approach the problem by trying to estimate the conditional probability distribution over multi-sets of semantic labels given a predicate and some features on that predicate and arguments. This model is then used to evaluate the likelihood of assigning a set of argument labels to a predicate-argument structure; the label assignment with maximum likelihood is the one that is selected by the model. This technique, however, can only condition on a limited number of features before the effects of data sparsity become overwhelming. Furthermore, like all other SRL systems, it relies on an automatic parse of the sentence to be labeled. Given that underlying parses may have errors, this system is not particularly robust to parser error. There are other techniques that attempt to overcome these limitations. A prominent one is the joint model proposed by (Toutanova, Haghighi, & Manning, 2008). Their log-linear model is trained in two steps. They first train a model to label each node in the parse tree independently of each other. The cost of a labeling of a sentence is simply the sum of the scores assigned to each label in the labeling. They then use this model to generate a *k*-best list of labelings for each sentence/parse tree, and use a conditional random field model (CRF) to rerank the labelings according to non-local features and features that exploit the dependencies between labels. They report significant reductions in labeling error over their local baseline.

Unfortunately, applying one of these methods to the chemistry lab domain will present many challenges. The domains on which parsers and SRL systems are trained on likely diverge significantly from the chemistry lab domain, although determining this is ultimately an empirical question. Lexically, we know the domains are rather different, since newswire is typically the source for the training corpora. One method of coping with this (if it turns out that off-the-shelf systems are too error-prone to construct symbolic representations of sentences) is to use a semi-supervised algorithm. This would require us to construct dependency structures on a subset of the corpus (again using an off-the-shelf syntactic/dependency parser), and labeling the nodes. We could then apply the method of (Furstenau & Lapata, 2009) to extract the symbolic predicate-argument information. Briefly, in their method, they infer a dependency structure on the unannotated sentences, and then compare those structures against the labeled structures in the seed corpus using a similarity measure. For each sentence, they compare the sentence's dependency structure against the structures in the seed corpus, select the seed structure that is 'most similar' to the candidate sentence, and project the labels from the seed onto the candidate. If we are careful about how we select the seed sentences from the chemistry lab corpus, this may be an effective way to construct sentence-level interpretations of the instructions. However, this method assumes that we have reasonably good dependency structures for the seed sentences. Given that we would be using off-the-shelf parsers, domain mismatch would likely be a concern as well.

To my knowledge, there is only one open source semantic role labeling system that has been released, SwiRL (http://www.surdeanu.name/mihai/swirl/), which uses AdaBoost to parameterize a model.

Coreference resolution

Before we construct a symbolic representation of each sentence, we need to decide what real-world objects each noun phrase (which will act as the arguments to the predicate) resolve to. For instance, if we have an instruction "Heat the test tube using the Bunsen burner", with the associated semantic representation HEAT(TEST_TUBE, BUNSEN_BURNER), we would need to resolve which test tube and which Bunsen burner we are referencing. Furthermore, if that instruction were followed by another saying "Brush it over the Bunsen burner periodically," the co-reference resolution step would need to resolve "it" as a reference to the test tube. There are two types of references: references that refer to a previously seen noun phrase (anaphoric references), and references that refer to a noun-phrase yet to occur (cataphoric references). Much work has been done on coreference resolution over the past 15 years or so. Most of the successful approaches have been through applying supervised machine learning algorithms to learn a model that resolves references based on the surrounding features. (Soon, Ng, & Lim, 2001) construct decision tree classifiers to classify whether a pair of noun phrases refer to the same referent based on vectors of features, including how many sentences apart the two noun phrases are, whether either of them are pronouns, whether the strings are identical, and whether they are persons. Others have augmented this feature set, including (Ng, Shallow semantics for coreference resolution, 2007) who added semantic type label features (not semantic role labels) found in the annotated ACE corpus, and (Ponzetto & Strube, 2006) who added semantic role label features. There has also been some work on unsupervised methods for coreference resolution, including (Haghighi & Klein, 2007) and (Ng, 2008). The most feasible solution to resolving coreferences in the text is to use the BART (http://bart-anaphora.org/) (Versley, et al., 2008) coreference resolution system. Given that its built in model was trained in a supervised manner from data, it is likely there will be domain mismatch, and so the quality of the references may be poor.

Constructing semantic predicates

Once we have a syntactic or dependency structure for a sentence, identified the predicates and labeled the arguments, labeled the types of objects to manipulate, and resolved the references, we need to construct the symbolic representations of the semantics that an inference engine will use for reasoning. This is not a common task in the NLP community and I know of no complete system that will construct these kinds of general semantic representations, and to my knowledge, there has not been much research in this area. If the goal is to automate sentence-level interpretation, then likely the best approach would be to construct a set of rules to recursively construct the predicate structures from the syntactic and/or dependency structures of the sentence. Although this will be error-prone, especially given a degree of domain mismatch between the data the parser is trained on and the chemistry lab corpus, we may be able to engineer rules that will be useful, especially if they don't need to generalize beyond a few chemistry lab documents.

It must be noted that due to the steps required to automatically interpret sentences, this pipeline is highly error prone, especially given domain mismatch issues. If the number of labs to interpret is small, and the goal of the study is on the role of deep domain knowledge to reason about the labs, it may be worthwhile to manually construct the semantic predicates from the sentences. It is my prediction that off-the-shelf systems trained on newswire (i.e. standard training data) will not lead to correct, informative semantic representations.

Analysis of Labs

In the section, we annotate each instruction and informational statement in two labs representative of the high-school chemistry lab corpus(Lomax, Dillner, & Streib), (Jircitano). For each sentence, we will discuss sentence-level and document level interpretation issues, discuss what deep domain knowledge can be extracted from the sentences, and discuss what deep domain knowledge could be leveraged to help interpret. In the section below, each distinct instruction or statement is analyzed, in regard to both sentence-level interpretation, and document-level interpretation. Each instruction annotation has the following format:

Step:

Instruction

Sentence-level interpretation issues:

Feasibility of constructing symbolic representation using a system trained in a supervised manner

Is there deep domain knowledge that we can extract from this?

Is there domain knowledge that would be useful for interpretation?

Document-level interpretation issues:

Is there domain knowledge that would be useful for interpretation?

Identification of whether the relevant domain knowledge is extractable from corpus

We assume that the interpretation pipeline is as follows:

1. Construct a syntactic/dependency parse for the sentence
2. Apply rules to identify the entities (equipment, chemical, property)
3. Run an SRL system to label the predicates and arguments
4. Resolve the references of noun phrases
5. Apply sequence of rules to convert labeled tree nodes into semantic predicates.

Trying to gauge the feasibility of interpreting each statement is difficult a priori. Ultimately, the feasibility is an empirical question that can be answered only with experimentation. The main sticking point is that if we take off-the-shelf systems trained on the standard corpora (i.e. Penn Treebank for parsers, PropBank, VerbNet, and FrameNet for SRL systems), there will be a domain mismatch between the training data and the test data (i.e. the features that the parsers and SRL systems will be trained on, namely, those extracted from the standard corpora, may not be found in the labs, and so the systems will make mistakes since the features they are trained to use for classification would not be found). Since it is difficult to estimate a priori how well off-the-shelf tools would work, in order to evaluate the interpretability of each sentence we need to make some assumptions. A large domain mismatch between the training and testing corpora would cause very noisy and incorrect annotations, which would prevent us from constructing semantic representations of many of the sentences. In order to be able to evaluate the role of deep domain knowledge in the text interpretation, we have to assume that domain mismatch will not be much of a problem. In analyzing the following text we assume that our off-the-shelf NLP systems can cope with the possible domain mismatch, and that we have some set of rules that can construct semantic predicate structures for reasoning from the syntax/dependency parse trees.

We should note that in the analysis of the labs, we found that the first lab will likely be easier to interpret due to simpler sentence structures, fewer parenthetical, and more declarative or imperative statements than in the second lab.

Lab 1: The Decomposition of Potassium Chlorate

Step 1

Record the atmospheric pressure from the laboratory barometer

Sentence-level:

Feasibility: strong – two roles, a theme (atmospheric pressure) and an instrument (laboratory barometer). Predicate "record" is found in PropBank and FrameNet, although PropBank does not contain the sense in which it is found here (FrameNet does)

Atmospheric pressure is recordable using a barometer

Domain knowledge not useful for interpretation.

Document-level:

In regards to the order in which instructions are executed, it would be useful to know that atmospheric pressure is constant in a laboratory (and so is not dependent on the time)

Relevant domain knowledge not extractable from the document

Step 2

Caution: $KClO_3$ is a very strong oxidizing agent

Sentence-level:

Feasibility: strong – simple declarative statement. Could possibly be processed by just a parser, without labeling the roles.

Can extract domain knowledge that $KClO_3$ is a strong oxidizing agent

Domain knowledge not useful in disambiguating.

Make certain you place the lid back on the bottle containing the $KClO_3$ after you obtain your sample.

Sentence-level:

Feasibility: moderate – complex sentence structure, containing instruction and constraint. Susceptible to parser error. Two predicates, so it could try to link arguments to the incorrect predicate. Both predicates exist in PropBank and FrameNet, so assuming correct parse, it could probably label arguments correctly.

Domain Knowledge applicability: Perhaps that one should put lids on bottles after use. Currently, that is too generalizable to infer automatically.

Domain knowledge that $KClO_3$ is a strong oxidizing agent would be a helpful constraint

Document-level:

Contains forward reference to future instruction. This constraint needs to be taken into account when processing future instructions.

Relevant domain knowledge not extractable from the document.

Do not let this substance contact paper or the rubber stopper in the test tube of the apparatus

Sentence-level:

Feasibility: moderate – PropBank and FrameNet do not have "contact" as a predicate where the sense is the same as it is used here (i.e. touch). It is possible that the

location argument ("in the test tube of the apparatus") will not be identified. The other predicates should be identifiable though.
No domain knowledge extractable. The reason for this instruction is not clear from just the lab.
It would be useful to have a model of the apparatus in the domain ontology to help resolve references.

Document-level:

Contains forward reference to future instruction. Needs to be taken into account when processing future instruction
Relevant domain knowledge not extractable from the document.

Step 3

Record the weight of the beaker on a top-loading balance

Sentence-level:

Feasibility: strong – ProbBank and FrameNet contain "record" as a predicate.
Contains three arguments, a hidden agent, a theme, and an instrument.
We should be able to extract that top-loading balances are used to measure weight
It would be useful to know that initially, the beaker is empty. The coreference resolution system should be able to resolve "beaker" and "balance" without too much difficulty.

Document-level:

Looking at instructions further down the line it would be useful to know that the beaker will no longer be empty, so it should be weighed before anything is put in it.
Relevant domain knowledge should be extractable, if we can reason that after we put something in the beaker, it is no longer empty

Assemble the apparatus as shown in Figure 1

Feasibility: impossible using NLP tools. No known NLP tools to interpret diagrams.

Step 4

Fill the Erlenmeyer flask with distilled water, so that the level of the water is about 1 inch below the short glass tube.

Sentence-level:

Feasibility: weak – complex sentence structure, with an instruction and a constraint.
There are two predicates in the sentence. The first (corresponding to the instruction) should be easy to extract; three simple roles, a hidden agent, a destination (the flask), and a patient (the water). The second predicate would be difficult to extract automatically, since we need to represent two concepts: the meaning of "1 inch below the short glass tube", and the water level conforming to that restriction. The coreference resolution system should be able to resolve "Erlenmeyer flask" and "short glass tube," but will have difficulty with water since water is not a distinct item.
Instruction causes a change in state which future instructions need to know about, so it should be added to the domain knowledge bank

It would be useful to have a theory understanding the distance between two objects (i.e. the water level and the short glass tube). It would also be useful to have a representation of "water level." Finally, a representation of the apparatus in the domain ontology may be useful for resolving references.

Document level:

Should know that the flask is initially empty

It would be useful to have predicates stating that the initial state of containers is empty

Open the pinch clamp.

Sentence-level:

Feasibility: strong – simple instruction, predicate in PropBank, SRL system should be able to interpret this accurately. Coreference resolution system should have no problem with "pinch clamp"

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

A representation of the apparatus in the domain ontology may be of use.

Document-level:

This is a prerequisite for future steps, so the change of state of the apparatus (i.e. that the pinch clamp is open) should be recorded.

This does not rely on the Erlenmeyer flask being filled. This step can be done in any order with respect to the previous and next instruction

Remove the stopper from the test tube.

Sentence-level:

Feasibility: strong – simple instruction, predicate in PropBank, SRL system should be able to label the roles correctly, and the coreference resolution should have no problem with "stopper" and "test tube"

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

Representation of apparatus in domain ontology would be useful

Document-level:

This is a prerequisite for future steps

This does not rely on the Erlenmeyer flask being filled or the pinch clamp being open. This step can be done in any order with respect to the two previous instructions.

Use a suction bulb to force air through the glass tube (test-tube end) until the rubber tube is filled with water.

Sentence-level:

Feasibility: moderate/weak – instruction with a constraint, two predicates in sentence. The first one (force) is found in PropBank, so an SRL system should be able to label the roles (although the parenthetical may make things more complicated). The constraint ("the rubber tube is filled with water") would likely be difficult to interpret automatically. The coreference resolution system may be able to resolve “bulb” “glass tube” and “rubber tube” but having the apparatus in the domain ontology would be better.

From this sentence, we could potentially extract the domain knowledge that blowing air into a closed system displaces the water. Automatically inferring that though is beyond the capabilities of the NLP state-of-the-art – an inference engine would need to reason this from the semantics

No domain knowledge needed

Document-level:

This relies on the previous three steps being completed (i.e. prerequisites are the Erlenmeyer flask being filled, the pinch clamp open, the stopper removed from the test tube)

Requisite domain knowledge extractable from correct interpretation of previous steps.

Allow a little water to enter the beaker to about 2 inches.

Sentence-level:

Feasibility: moderate/weak – instruction and constraint. “Enter” is found with the correct sense in PropBank. The constraint “to about 2 inches” may cause problems though because constraints are not listed as arguments to the predicate in PropBank. Furthermore, “to about 2 inches” does not give a reference point (2 inches with respect to what?). I don’t believe this can be automatically disambiguated. This is ambiguous for humans as well (although it likely refers to “2 inches from the bottom of the beaker.”) The coreference resolution system should be able to resolve “beaker” but will likely have trouble with “water”

The change of state in the experimental setup should be noted (since this step is a prerequisite to future ones).

Domain knowledge may be useful to disambiguate to “about 2 inches.”

Document-level:

It is not clear how we are to allow the water to enter the beaker. It would be useful to have a theory representing the effects of pressure. This predicate would need to be resolved with the predicate of the previous instruction (since using the suction bulb is what should cause the water to enter the beaker).

Requisite domain knowledge extractable from correct interpretation of previous steps.

Close the pinch clamp near the end of the tubing where the water will exit.

Sentence-level:

Feasibility: moderate – relatively simple structure, PropBank contains the predicate and the correct sense. The coreference resolution module would need to resolve “the tubing where the water will exit.” This requires understanding of the system as a whole, and it is unlikely that the coreference resolution system will be able to resolve this on its own.

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

It would be useful for coreference resolution to understand the steps taken previously, to have a theory about the effects of pressure on water, and to understand that if water is displaced it must go somewhere (and through something). It would also be useful to have a good internal representation of the apparatus, to help resolve the references.

Document-level:

Relies on experiment state after the previous steps

Requisite domain knowledge extractable from correct interpretation of previous steps.

Step 5

Make sure that the test tube is clean and dry.

Sentence-level:

Feasibility: moderate – "Make sure" is an idiomatic expression that does not appear in PropBank. If we were to use, say WordNet, to replace "make sure" with verify, then this should be easily interpretable. The co-reference resolution system would need to resolve which test tube the argument is referring to.

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

Having an understanding of the apparatus and the experimental set up may help disambiguate the reference of "test tube."

Document-level:

This is a prerequisite for next few instructions.

Take the test tube and a clean, empty, dry 400 mL beaker to the top-loading balances.

Sentence-level:

Feasibility: strong – three arguments, predicate found in PropBank. Coreference resolution step should be able to resolve "test tube" to that in the previous instruction

No domain knowledge extractable

Domain knowledge not useful in disambiguating

Document-level:

The test tube must be clean and dry. Needs to know the state after the previous step. Should be done before the test tube and beaker are used in an experiment.

Requisite domain knowledge may be extracted by considering future steps.

Take the empty beaker, add the test-tube and record its mass.

Sentence-level:

Feasibility: moderate – This instruction is really asking to record the mass of the empty beaker together with the test tube. An SRL system probably could not extract that information from this sentence in its current form. If it were paraphrased, we may have more success. The equipment should be easily resolvable based on previous instructions.

No domain knowledge extractable

It would be useful to know that top-loading balances are used to measure mass (which we can get from this document).

Document-level:

Must be done before test tube and beaker are used in an experiment.

Take the beaker and test-tube.

Sentence-level:

Feasibility: strong – simple instruction, two roles, predicate in PropBank, previously seen noun phrase arguments.
No domain knowledge extractable.
No domain knowledge useful.

Step 6

Carefully add a small amount of $KClO_3$ to the test tube.

Sentence-level:

Feasibility: strong – simple instruction, two explicit roles, one hidden. "Test tube" should be easy to resolve given previous instructions. Coreference system should be able to resolve "test tube" and $KClO_3$
Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions.
It would be useful to have a definition of "small amount", which we get in the next sentence (1.0 g)

Document-level:

Domain knowledge critical, since there is a constraint from step 2.

Continue to add $KClO_3$ until you have about 1.0 g of $KClO_3$ in the test tube.

Sentence-level:

Feasibility: moderate – the sentence structure is complex. Two of the arguments of "add" are missing (the agent and the item being added to), and so the traces of the missing arguments (especially the item being added to) must be resolved to be the test tube. Also, it contains a constraint that would be difficult to automatically extract.
The knowledge that the test tube contains 1.0g of $KClO_3$ is extractable
Knowing that the test-tube is a container (as perhaps identified by the named entity recognition step) may be helpful in determining the third argument of "add"

Document-level:

Considering the previous instruction that the potassium chlorate should be added to the test tube may be useful in disambiguating this instruction.
Domain knowledge is critical here considering the constraint from step 2.

A sample in the range of 0.9 g to 1.1 g of $KClO_3$ will work.

Sentence-level:

Feasibility: weak – This is a highly idiomatic expression.

Record the mass to 0.01 g

Sentence-level:

Feasibility: moderate – The sentence structure is simple, but there is some ambiguity in resolving what "mass" refers to. Mass is a property, and so it needs to be resolved to the entity of which it is a property (namely the test tube with $KClO_3$). Also, it is not clear that "to 0.01 g" means to a certain degree of accuracy. That cannot be automatically interpreted.

No domain knowledge extractable

It would be useful to know that top-loading balances are used to record mass, and it would be useful to have an idea of what measuring something to a particular degree of accuracy means.

Document-level:

Must follow previous steps.

Step 7

Clamp the test-tube to the ringstand and stopper the test tube.

Sentence-level:

Feasibility: strong – Simple instruction, two predicates, clamp found in PropBank, stopper found in VerbNet. Coreference of "test tube" should be easy to resolve given previous sentences.

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

No domain knowledge needed to interpret

Document-level:

No domain knowledge needed, other than that this needs to follow the previous steps.

Step 8

Open the pinch clamp.

Sentence-level:

Feasibility: strong – Simple instruction, single predicate, two arguments.

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

Having a good representation of the apparatus would help resolve entity instances (i.e. the pinch clamp)

Document-level:

No domain knowledge needed, sentence can be interpreted on its own.

Lift the beaker with your hands until the water level in the beaker is equal to the water level in the Erlenmeyer flask.

Sentence-level:

Feasibility: weak – Instruction with constraint. The instruction should be straightforward to interpret, but the constraint is not. There are few examples of "equal" as a predicate in PropBank, and is not found in FrameNet or VerbNet. Significant domain mismatch between training data resources and this corpus.

If we could interpret this, and had domain knowledge about the effects of pressure on liquids, we may be able to generalize that lifting the beaker creates a difference in pressure, which causes the liquid to flow from the beaker into the flask.

Document-level:

Requires knowledge of the effects of the previous steps on the state of the experiment and the apparatus

When the water levels are equal, have your partner close the pinch clamp.

Sentence-level:

Feasibility: moderate – Constraint and instruction. The instruction should be reasonably simple to interpret. PropBank has the proper sense of the predicate "have," as well as the proper sense of the predicate "close." The SRL system would need to interpret both the dependent clause "close the pinch clamp" and the outer clause "have your partner X."

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

Useful to have representation of the apparatus to help resolve the reference to the pinch clamp.

Document-level:

Must resolve this symbolic representation of this constraint with the instruction to equalize in the previous step

This equalizing process will ensure that the pressure acting on the water in the beaker (atmosphere) is equal to the pressure acting on the water in the flask.

Sentence-level:

Feasibility: moderate/weak – complex sentence structure (dependent clause), so there are two predicate/argument pairs to extract, one that the pressure acting on the beaker water is equal to the pressure acting on the flask water; and the other that it is the equalizing process of the last instruction that ensures this equality.

Resulting symbolic representation of this statement can be added to the domain knowledge base

Domain knowledge not necessary for interpretation

Document-level:

Relies on correct interpretation of previous instruction in order to resolve "This equalizing process" with the instructions on how to equalize.

Step 9

Empty the beaker but do not dry it.

Sentence-level:

Feasibility: strong – simple instruction with constraint. The instruction should be easy to interpret, and the coreference resolution system should be able to resolve the beaker to previous mentions of it, and it should be able to resolve the pronominal "it" to "the beaker." The constraint might not be properly labeled by the SRL system since it's a dependent clause.

No domain knowledge extractable

No domain knowledge needed

Document-level:

Need to know the result of the previous step, namely, that there is water in the beaker.

The volume of the water drops that remain in the beaker will be roughly equal to the volume that will remain after the displaced water is poured into a graduated cylinder for measurement

Sentence-level:

Feasibility: weak – The SRL system should be able to label the arguments correctly, but it will be difficult to construct symbolic representations of the arguments themselves. “The volume of the water drops that remain in the beaker” could be resolvable to the symbolic representation of the result of the previous step (i.e. emptying the beaker but not drying it), but this requires deep semantic understanding of the effects of emptying the beaker, and no coreference system utilizes those kind of features. Then, “the volume that will remain after the displaced water is poured into a graduated cylinder for measurement” references a future step. This cannot be resolved until we interpret that future step.

Resulting symbolic representation can be added to domain knowledge bank

Domain knowledge could help in interpretation.

Document-level:

This is an informational statement, it is not an instruction or step. Therefore, we can interpret other steps first, and that would help us interpret this informational statement.

Step 10

Place the glass tube (connected to the hose) back into the beaker.

Sentence-level:

Feasibility: moderate – The SRL system should be able to label the arguments correctly; common predicate in PropBank with correct sense. Resolving the arguments may be more difficult, but having a representation of the apparatus setup should help (if we can map the surface forms of the arguments in this sentence to the same semantic representation extracted from the apparatus diagram).

No domain knowledge extractable

Having domain knowledge of apparatus set up should help resolve the references.

Document-level:

Instruction needs to be executed after the previous ones, since at this point the beaker has to have been emptied.

Make certain the pinch clamp is open

Sentence-level:

Feasibility: weak – “Make certain” is an idiomatic expression not found in PropBank. It would need to be paraphrased as “verify” before application of SRL system. SRL system would need to label the dependent clause too (i.e. construct a symbolic representation of the pinch clamp being open). In fact, it would be easier to construct a symbolic representation if this were paraphrased as “Open the pinch clamp.”

No domain knowledge to extract

Domain knowledge not useful in interpretation

Document-level:

Not really dependent on other instructions for interpretation.

Caution: If the clamp is not opened at this point, the build-up of gas during heating could cause an explosion, although it is more likely that a stopper would be forced to loosen.

Sentence-level:

Feasibility: weak – Complex sentence structure, susceptible to parser error. Many predicates, refers to a future instruction. It is very unlikely that an SRL system would be able to construct such a complex symbolic representation.

Should be able to extract knowledge that the increase in pressure during heating could cause an explosion or loosen the stopper. But first, we need a symbolic representation of this information.

A set of predicates about the apparatus should help resolve the argument NPs in the statement.

Document-level:

This is a constraint that applies when we're heating the test tube. This is a critical constraint to apply to step 11.

Also, make certain that the longer glass rod is not touching the bottom of the Erlenmeyer flask.

Sentence-level:

Same issues with the previous sentence apply to this one.

This would also result in a closed system and an explosion could result.

Step 11

Heat the test tube.

Sentence-level:

Feasibility: strong – This is a very simple instruction that is not very ambiguous. The SRL system should have no difficulty in finding the arguments and labeling them.

Instruction implies a change in state in the experimental apparatus which is a prerequisite for future instructions

It may be useful to have a representation of the apparatus diagram to help resolve the reference "test tube," although the coreference resolution system should be able to get that already

Document-level:

Prerequisite is that the test tube have a solid in it, so the ordering of this instruction is important.

The solid will melt, oxygen will be evolved, and water from the flask will be displaced into the beaker.

Sentence-level:

Feasibility: weak – Statement describe the change in state due to heating the test tube. There are lots of challenges in interpreting this, including that there are several predicates (in conjunction) in this sentence, that there is ambiguity in resolving the reference of "the solid," and that although all the predicates in the sentence appear in PropBank and FrameNet, their senses are not commonly seen in those corpora.

The effects of heating potassium chlorate are described here, and would be a good addition to the domain knowledge bank.
It would be useful to understand the chemical composition of $KClO_3$, so that when we say "Oxygen will be evolved," we can resolve the oxygen to that which was in the potassium chlorate molecule.

Document-level:

This statement needs to be interpreted in the context of the previous instruction, namely that these effects are due to the heating of the test tube. As this is a descriptive statement, not an instruction, the truth of this statement is independent of the order in which the instructions are executed (the antecedents are simply that the test tube contain potassium chlorate).

Be cautious at first and brush the flame over the test tube.

Sentence-level:

Feasibility: strong – simple constraint on how to heat the test tube. An SRL system should be able to isolate the predicate "brush" from the sentence and label the arguments ("flame" and "over the test tube").
No domain knowledge to extract
It would be useful to know that the Bunsen burner emits a flame, so that we can then resolve "flame" to a property of the Bunsen burner.

Document-level:

After a few minutes when the liquid solidifies, the test tube can be heated more strongly.

Sentence-level:

Feasibility: moderate – the SRL system should be able to identify that "heat" is the predicate with a single argument "test tube." There is a constraint, namely the condition that when the liquid solidifies. Automatically interpreting the constraint "After a few minutes when the liquid solidifies" will likely not be possible, with the SRL system.
Could extract knowledge that liquid potassium chlorate solidifies once the oxygen is evolved. This may be inferred using an inference engine once we have symbolic representations of the predicates.
It would be useful to know that when potassium chlorate is heated, it liquefies (domain knowledge we can extract from the two previous sentences); or rather, that any solid melting implies that the solid is now a liquid.

One gram of $KClO_3$ reactant should cause the displacement of between 250 and 300 mL of water.

Sentence-level:

Feasibility: strong – the SRL system should be able to process the "displacement" predicate and its arguments, and we could extract a symbolic representation of the arguments by considering the syntactic subtrees of those arguments. It would be difficult to automatically resolve "water" though, since it is not clear whether this refers to the water in the flask or the remainder that is in the beaker.

This statement is deep domain knowledge that could be extracted for use in reasoning about the lab
Semantic representation of the apparatus would be helpful in resolving "water"

Step 12

Heat the solid thoroughly until no more gas is evolved.

Sentence-level:

Feasibility: moderate – instruction with constraint. The SRL system should be able to identify the "heat" predicate with the "solid" argument, but it may have trouble with the dependent clause (i.e. the constraint). It is unlikely that coreference resolution system would be able to resolve "solid"
Could extract knowledge that by heating the potassium chlorate for a period of time will cause all of the oxygen to evolve.
It would be useful to understand the chemical composition of potassium chlorate.

The contents of the test tube will solidify, since the melting point of the product is greater than that of $KClO_3$

Sentence-level:

Feasibility: moderate– observational statement with an explanation. Two predicate/argument pairs, but reasonably simple sentence structure, so we should be able to parse it with little difficulty. The coreference resolution system should be able to resolve "test tube" and $KClO_3$, but "melting point" and "product" will be difficult to interpret.
The explanation could be added to the domain ontology
It would be useful to have an understanding of the "melting point" property in the domain ontology, as well as have an understanding of the chemical reaction that lead to the transformation of $KClO_3$ into a different compound.

Step 13

Turn off the flame and allow the system to come back to room temperature.

Sentence-level:

Feasibility: strong – the SRL system should be able to represent the instruction "turn off the flame" without much difficulty.
No domain knowledge to extract
We would need to know that the Bunsen burner is the source of the flame in order to resolve the reference.

Allow five minutes for this process.

Step 14

As in step 8, equalize the water levels (this may require lifting the Erlenmeyer flask) and close the clamp

It is unclear how to process this instruction. It may be better to replace this with text from step 8

Step 15

Remove the tube from the beaker.

Sentence-level:

Feasibility: strong – simple instruction, the SRL system should be able to analyze it correctly.
No domain knowledge to extract.
It would be useful to have representation of the apparatus to help resolve the arguments

Record the mass of the beaker plus the water on a top-loading balance.

Sentence-level:

Feasibility: strong – simple instruction, the SRL system should be able to analyze it correctly. We may need to use the subtree to construct a representation of "beaker plus water"
No domain knowledge to extract
It would be useful to know that top-loading balances are used to measure mass.

Use Table I to determine the volume of water displaced

Uninterpretable

Step 16

Measure the temperature of the water to the nearest degree.

Sentence-level:

Feasibility: strong – simple instruction, the SRL system should be able to analyze it,
No domain knowledge to extract
Knowledge that thermometers are used to measure temperature could be used in refining the symbolic representation of the instruction.

Assume this is the temperature of the gas.

Sentence-level:

Feasibility: strong – simple instruction, the SRL system should be able to analyze it.
Resolving "this" may be tricky though (pronoun resolution).
Can be added to domain knowledge bank.
No domain knowledge necessary

Determine the appropriate vapor pressure of water from Table II

Uninterpretable

Step 17

Obtain the mass of the test-tube and its contents.

Sentence-level:

Feasibility: strong – simple instruction, the SRL system should be able to analyze it
No domain knowledge to extract
Knowledge that balances are used to measure mass should be used to refine the symbolic representation of the instruction

Calculate and record the mass of the product.

Sentence-level:

Feasibility: weak – two predicates, calculate and record. Calculate has two arguments, both hidden (the theme is a trace of "the mass of the product.") The SRL system would have to be able to label two predicates, and it is unlikely it would be able to do so

No domain knowledge to extract

Domain knowledge not useful

Lab 2: Studying the Decomposition of Potassium Chlorate

Part A: Verifying the Identity of the Gas Product

Step 1

Observe the correct usage of the Bunsen burner, as demonstrated by your instructor.

Not a relevant instruction

Make any special notes for future reference.

Not a relevant instruction

Step 2

Obtain a small, dry test tube and connect the stopper assembly to it (stopper with single glass tube and tubing).

Sentence-level:

Feasibility: moderate – two predicates, "obtain" and "connect." "Obtain" should be interpretable, as there are two arguments, a hidden one and the test tube. "Connect" would be more difficult, since "stopper assembly" would likely not be in any training corpus. Furthermore, we would have to resolve the pronoun "it" to test tube, and the system may not be able to cope with the parenthetical.

Can extract that once this instruction is processed, the state of the apparatus would change

No domain knowledge needed on sentence-level

Document-level:

It would be useful to have a representation of the stopper assembly in the knowledge bank, similar to the diagram of the apparatus in the previous lab.

Place the end of the tubing into a small plastic bin containing water.

Sentence-level:

Feasibility: moderate – This has very simple structure, and the SRL system/parser should be able to construct a symbolic representation (single predicate "place", three arguments – hidden agent, theme "end of the tubing", destination "small plastic bin containing water." The difficulty in interpreting this involves resolving "the end of the tubing" on the document-level.

No domain knowledge to extract

No domain knowledge useful for sentence-level interpretation

Document-level:

It would be useful to have a representation of the assembly so that we can resolve "end of the tubing."

Heat the empty test tube for a few seconds, holding the tube with test tube holders.

Sentence-level:

Feasibility: moderate/weak – instruction with constraint; two predicates ("heat" and "holding"). "Heat" has two arguments, hidden agent and a theme (empty test tube). "holding" has three arguments, a hidden agent (a trace), a theme (the tube) and an instrument (test tube holders). The "tube" in the constraint needs to be resolved to the test tube from the first instruction of the step.

We could extract the constraint to add do the domain knowledge bank that when test tubes are heated, they should be handled with test tube holders.
It would be useful to know that test tubes are to be heated with Bunsen burners so that we can construct a proper predicate (since it would be most useful for "heat" to have an instrument argument).

What do you observe right away?
Not interpretable

What gas is escaping from the tube?
Not interpretable

After heating, remove the stopper to prevent water from being pulled into the cooling test tube.
Sentence-level:
Feasibility: weak – instruction with a constraint/explanation. Two predicates, "remove" (with two arguments, hidden agent and "stopper" as the theme), "prevent" (with a single argument, an action, a second predication); and "pull," with two arguments; a patient, the water, and the direction (into the cooling test tube). This is a very complex sentence structure, and it is unlikely that an SRL system would be able to get this right. The best bet to interpret this would be to use a dependency parser and then use a set of deterministic rules to construct the semantic representation.
From this, the inference engine may be able to reason that when in a closed system such as the apparatus, as the test tube cools, water will flow into the test tube. This could be added to the domain knowledge bank. The fact that this is due to a decrease in pressure in the test tube cannot yet be deduced from this lab alone.
Domain knowledge not needed for interpretation.
Document-level:
This step is only applicable after we've heated the test tube in the previous step.

Step 3
With a weighing boat, add about 0.3 g of potassium chlorate to a small, dry test tube (use the tube from step 2 when it is cooled).
Sentence-level:
Feasibility: moderate/weak – instruction with parenthetical constraint. Two predicates: "add," that takes four arguments: a hidden agent, an instrument (the weighing boat), an agent (0.3 g of potassium chlorate) and a patient (the test tube), and "use", taking two arguments (hidden agent [trace] and patient [the tube from step 2]). The SRL system should be able to label the arguments for "add", although the fact that the instrument is not in the canonical location may be problematic. The coreference resolution system should be able to resolve the "small dry test tube." Given that this sentence contains many lexical items that specific to the chemistry lab domain, it is unlikely that we will get a good syntactic/dependency parse of this sentence, and so the labelings will be suspect.
Can add knowledge of the usage of the weighing boat to the domain ontology

Domain knowledge not needed for interpretation

With a spatula, add a small amount of the catalyst manganese (IV) oxide to the tube (about the size of a "pinch of salt" or fraction of the tip of the spatula).

Sentence-level:

Feasibility: moderate – very similar structure to the previous sentence, same comments apply.

Be careful, since MnO_2 stains hands and clothing.

Sentence-level:

Feasibility: strong – simple sentence structure, two predicates ("be careful" will likely be ignored since it's not in FrameNet/PropBank/VerbNet) and "stains" (two arguments, agent (manganese) and patient (hands and clothing). MnO_2 needs to be resolved to Manganese (which the coreference resolution system will be unable to do), and the patient does not resolve to anything since it is a generic statement. This constraint applies throughout the lab, so this should be added to the domain knowledge bank.

We can only resolve MnO_2 to Manganese only if it's in the domain ontology.

Though no exact mass of catalyst is required, the mixture should appear light grey after gently tapping the test tube to mix the contents uniformly.

Sentence-level:

Feasibility: weak – the structure of this sentence is very complex, and it is unclear which of the words are the predicates. The relevant part of the sentence is "the mixture should appear light grey after gently tapping the test tube to mix the contents uniformly." It is a statement with a contextualizing constraint (i.e. the mixture will appear light grey only when the statement "after gently tapping the test tube to mix the contents uniformly" is satisfied). The first statement has one predicate, "appear," that takes two arguments – the mixture (agent) and light grey (predicative argument). The constraint has a predicate, "tapping", with three roles – a hidden agent, a patient (the test tube) and a predicative argument (to mix the contents uniformly). Given that both statements have predicative arguments, it is unlikely the SRL system will be able to label them. It may be better to construct a syntactic analysis of this sentence and then apply rules to convert it to a semantic representation.

Add a small, loose layer of glass wool to the top of the tube which will allow gases to escape but keep solids from splattering onto the stopper.

Sentence-level:

Feasibility: weak – very complex sentence structure, same interpretative problems as in the previous instruction.

Part B: Quantitative Study of the Decomposition Reaction

Step 1

Fill a 600 mL beaker ½ full with distilled water.

Sentence-level:

Feasibility: strong – simple instruction, single predicate, three arguments (one hidden agent, one patient (beaker), and one theme (water)). Coreference resolution should be simple (since this is the first mention of this piece of equipment and distilled water).

Domain knowledge not needed for disambiguating

Knowledge that the beaker has roughly 300 mL of distilled water will be useful in interpreting future instructions (on the document level)

Document-level:

No additional interpretation on document level needed

Step 2

Fill the large gas collection bottle about ¾ full with distilled water.

Sentence-level:

Feasibility: strong – simple instruction, single predicate, three arguments.

Coreference resolution should be simple, although coreference resolution system will likely label the distilled water as being a reference to the distilled water mentioned in the previous instruction. They probably should not be coreferenced, since it’s not the same “instance” of distilled water.

Domain knowledge could perhaps be used to determine that the distilled water mentioned here is not the same as the distilled water in the previous instruction – namely, that water is a more-or-less continuous supply and that we can divide it endlessly.

The change of state of the gas collection bottle (that it is now ¾ full of water) will be useful in interpreting future instructions.

Document-level:

The next constraint would be useful in giving sanity check to the agent interpreting the text. This instruction and the previous can be executed in any order relative to each other. (i.e. domain knowledge that the two containers are independent may be useful).

Relevant domain knowledge could be extracted from next instruction.

With the stopper inserted, the water level should be slightly below the short tube inside the bottle (the long tube will be immersed).

Sentence-level:

Feasibility: weak – a constraint on one of the previous instructions. Complex sentence structure, predicate is an identity (i.e. “is”). It is unlikely an SRL system would be able to label this. Arguments (“With the stopper inserted”, “slightly below the short tube inside the bottle”) are both predicative. Also, there is an elliptical reference (it is not clear where the stopper should be inserted), so that will be difficult to resolve.

Domain knowledge not useful on the sentence level

Assuming we had a good symbolic representation of the semantics, this should be useful in executing the previous instruction, as it gives the agent interpreting the instructions a sanity check.

Document-level:

It would be useful to have a representation of the experimental apparatus in the domain ontology to interpret the statement.

Clamp the bottle to a ring stand.

Sentence-level:

Feasibility: moderate/weak – simple instruction, but predicate is not found in this sense in the training corpora (namely PropBank; in VerbNet, "clamp" is mapped to "tape" and its senses/arguments). Three arguments (one hidden, a theme, and a patient). Coreference resolution system would need to resolve the bottle to the gas-collection bottle two instructions ago, and ring-stand to a representation of the apparatus in the domain ontology.

Domain knowledge would be critical in resolving the reference to "ring stand"

Describes change in state of the experimental apparatus that would be critical for interpreting future instructions.

Document-level:

This must be done once the bottle has water in it, since it can't be filled when it's on the ring stand.

Relevant information not extractable from the lab.

Place the long tubing end (not the one with the stopper) into the beaker of water.

Sentence-level:

Feasibility: strong – simple instruction, predicate is found in training corpora (with the proper sense). Coreference resolution system should be able to correctly resolve "beaker of water." It will not be able to resolve "long tubing end" on the sentence-level.

Domain knowledge needed to resolve "long tubing end," so having a representation of the apparatus in the domain knowledge bank would be needed.

Describes change in state of the experimental apparatus that will be useful in interpreting future instructions.

Document-level:

Can only be done following the previous instructions. Requires that beaker contain water, and that gas collection bottle does as well.

Do not attach the pinch clamp yet.

Sentence-level:

Feasibility: strong – simple instruction, predicate in training corpus, two arguments (hidden agent, theme). Coreference resolution system will likely be unable to resolve "pinch clamp" without using domain knowledge

Domain knowledge needed to resolve "pinch clamp", so having a representation of the apparatus in the domain ontology would be useful.

Domain knowledge not extractable

Document-level:

Only applies until next instruction is interpreted.

Step 3

With a pipette bulb, slowly blow air into the stopper end of the tubing until the water fills the long glass tube and tubing at the other end (no bubbles left in the exit tube or tubing) making sure the tubing end remains under the beaker water level.

Sentence-level:

Feasibility: weak – Very complex sentence structure. A simple instruction with two constraints. There are several predicates in this sentence: "blow", taking four arguments (hidden agent, a theme (air), a patient (stopper end of the tubing), and an instrument (pipette bulb)). There are two constraints: "until the water fills the long glass tube and tubing" and "making sure the tubing end remains under the beaker water level." The SRL system will not be able to label these and the syntactic or dependency structures that a parser will assign them, even if they are correct, would likely be too complex to simply transform into a set of predicates/arguments. Furthermore, there are many arguments that we would need to map to instances, (pipette bulb, air, stopper, long glass tube, tubing, beaker water level). Interpreting this sentence is quite beyond the state-of-the-art.

Domain knowledge may be useful in resolving some of the references to parts of the experimental apparatus. Having a symbolic representation of the concept of water level may also be useful.

Attach the pinch clamp to the tubing as close to the beaker as possible.

Sentence-level:

Feasibility: moderate – simple sentence structure, single predicate "attach," which has three arguments (hidden agent, theme (pinch clamp), and patient (the tubing as close to the beaker as possible)). The SRL system should be able to label "clamp" and "tubing." Interpreting "as close to the beaker as possible" is a bit trickier, but likely the best way to do it would be to treat it as an NP, and resolve it to knowledge extracted from the apparatus diagram.

A representation of the apparatus in the domain ontology would be useful in resolving references.

Describes change in state of apparatus, needed for future instructions.

Document-level:

Negates the instruction to not attach the pinch clamp two instructions prior.

Make sure the bottle is still more than ½ filled with water.

Sentence-level:

Feasibility: moderate – "Make sure" is an idiomatic expression that does not appear in PropBank. We could use another ontology (e.g. WordNet) to replace "make sure with "verify." The coreference resolution system should be able to resolve "bottle." It may try to resolve "water" as well, which it probably shouldn't.

A representation of the apparatus in the domain ontology might be useful in help resolving the references if the coreference resolution algorithm is not able to.

No domain knowledge extractable.

Document-level:

This instruction is a constraint on the previous one, so the previous instruction may need this for interpretive purposes.

Step 4

Obtain your potassium chlorate unknown (mix of $KClO_3$ and an inert material) and record its unknown code.

Sentence-level:

Feasibility: moderate– two instructions, with two predicates ("obtain" and "record"). The parenthetical would need to be resolved to "potassium chlorate unknown," which would be difficult for a coreference resolution system to resolve. The useful information to extract is that the agent interpreting the instruction should now have a sample of potassium chlorate unknown in its possession, and that this unknown is a mix of potassium chlorate and inert material. This would be difficult for NLP tools to infer automatically.

It would be useful to know that $KClO_3$ is synonymous with potassium chlorate.

This step introduces a new entity to manipulate, the potassium chlorate unknown, that will be useful in resolving future instructions.

Document-level:

This instruction can be done at any time with respect to the previous instructions, so temporally it does not rely on any of the previous instructions.

Step 5

Obtain a large, dry test tube and measure its mass using an analytical balance.

Sentence-level:

Feasibility: strong – two instructions with two predicates ("obtain" and "measure"). Coreference resolution system should be able to resolve "its" to the dry test tube (and thus create symbolic representations of the arguments of the "measure" predicate).

Can extract knowledge that analytical balances are used to measure the mass of an object.

It would be useful to know that entities have a property called "mass."

Document-level:

This instruction can be done at any time with respect to previous instructions.

To weigh a test tube, use a small beaker to hold it up on the balance.

Sentence-level:

Feasibility: moderate/weak - this is an instructional statement, telling us how to use the balance to weigh the test tube. It is highly ambiguous, because it is not clear from considering this sentence alone that the test tube should be placed inside the beaker. It is unlikely that the SRL system would label the arguments correctly, because the predicate "use" has many senses that it's not particularly informative. It may be better to construct a syntactic or dependency parse and extract a symbolic representation from it. This sentence is a statement that is always true, regardless of what "test tube," "small beaker," and "balance" resolve to, although it is difficult to tell automatically from the context. Perhaps the fact that the determiners for "test

tube" and "small beakers" are indefinite articles may be useful features for deciding not to resolve them, but I don't know of any off-the-shelf system that would use them.
If we had a good symbolic representation for this statement, it would be useful to add to the domain ontology.
It would be useful to know that small beakers are containers (perhaps the NER system could identify this) in order to know to put the test tube in the beaker.

Document-level:

This statement is always true, so it will be applicable throughout the whole document.

Add 0.3 +/- 0.05 g of your unknown to a weighing boat (using a top-loading balance for this "pre-weight" is fine)

Sentence-level:

Feasibility: moderate/weak – two separate instructions. The first one has "add" as the predicate with a hidden agent, a theme (0.3 +/- 0.05 g of your unknown), and a patient (a weighing boat). The SRL system should be able to label the arguments correctly, but constructing internal semantic representations of the arguments may be tricky – "unknown" needs to be resolved to the potassium chlorate compound, and 0.3 g +/- 0.05 g is actually a separate step, since doing so requires measuring it and there is no explicit instruction to do so. The second instruction, that using a top-loading balance is also difficult, because it is not clear (even to me) what exactly they mean by 'this "pre-weight" is fine.'

Document-level:

Once this instruction is processed, the unknown compound will be in the weighing boat.

Transfer the unknown to your test tube.

Sentence-level:

Feasibility: strong – simple instruction, SRL system should label the arguments correctly. Coreference resolution system should be able to resolve the unknown to the previous instruction and the test tube to two instructions ago.
That the test tube now contains the sample of the unknown will be useful in interpreting instructions to follow.
In order to interpret this sentence properly, we would need to know where the unknown is, which is domain knowledge we can extract from the previous statement.

Use an analytical balance to obtain the mass of the test tube containing the unknown to the nearest 0.0001 g.

Sentence-level:

Feasibility: moderate – two predicates, the outer one "use" has three arguments, a hidden agent, an instrument (analytical balance) and a predicative argument ("to obtain the mass…"). If the SRL system can interpret more than one predicate in a sentence, then it should not have any difficulty in interpreting the test tube and mass arguments to "obtain". It is not clear, however, that the coreference resolution system will be able to resolve the test tube reference, and without domain knowledge, it will not be able to interpret "the nearest 0.0001 g."

Can extract knowledge that analytical balances can be used to measure mass.

It would be useful to have the concept of the values that the mass property of an object can take, and to understand the concepts of significant digits with respect to measurement

Document-level:

This can only be interpreted once the test tube contains the unknown, and before it is manipulated, so this step must be performed now relative to previous and future instructions.

Step 6

Add a small amount of MnO_2 to form a light grey uniform mixture after mixing.

Sentence-level:

Feasibility: weak – Two predicates, an outer one "add" and an inner one "form." The SRL system should be able to label the arguments to "add", but interpreting the inner constraint ("to form alight grey uniform mixture after mixing") will be difficult. Furthermore, there is a lot of ellipsis in this instruction – it is not explicit what is being mixed, and it's unclear to what we should add the small amount of MnO_2. It's unlikely that these missing arguments can be automatically inferred.

Once this is interpreted, we know that the test tube contains a mixture of potassium chlorate unknown and the catalyst, which is required for future instructions, so this can be added to the domain knowledge bank

Domain knowledge not helpful in interpreting.

Document-level:

N/A

Add a small, loose layer of glass wool to the test tube.

Sentence-level:

Feasibility: moderate – simple instruction, but the lexical items are rare (even in this corpus) so the features for labeling may not be very informative given the model.

The coreference resolution system should be able to resolve "test tube."

Once this is interpreted, we can add this to the domain knowledge bank for usage by following instructions.

Domain knowledge not needed in interpretation.

Document-level:

N/A

Step 7

Weigh the test tube and its contents to the nearest 0.0001 g.

Sentence-level:

Feasibility: moderate – simple instruction, SRL system should be able to label it correctly. Coreference resolution system should also have no problem resolving "test tube." Domain knowledge critical in interpreting this, though.

No domain knowledge to extract.

Domain knowledge critical in interpretation. We need to have knowledge of significant digits, we need to know that "weighing" something means measuring its mass, and we need to know that one uses an analytical balance to measure mass.

Document-level:

Must be done after the previous steps

This will be your mass before heating.

Sentence-level:

Feasibility: moderate – simple instruction, although the predicate is an identity ("will be"), so it may be difficult to interpret. Resolving "this" may be tricky as well, since there's no explicit mention of mass in the previous instruction.
It is not clear that this is a useful instruction, so whether domain knowledge is extractable is debatable.
Domain knowledge not needed for interpretation

Document-level:

N/A

Step 8

Attach the test tube to the stopper end of the tubing using a slight twisting motion to ensure a good seal.

Sentence-level:

Feasibility: moderate – instruction with constraint. The instruction should be reasonably simple to label ("attach" is the predicate, "test tube" is the theme, "stopper end of the tubing" is the patient, and "using a slight twisting…" is the means by which it is attached). Resolving the references is more difficult, and having knowledge of the apparatus should be useful. The constraint (i.e. the means) is much more difficult to interpret, since "slight twisting motion" cannot be resolved to anything, and it is not clear how to represent that symbolically. Furthermore, there is a hidden argument (ellipsis) in the argument "good seal" to the predicate "ensure," which will be difficult for a coreference resolution system to resolve.
Change of state in the experimental apparatus useful for future instructions.
It would be useful to have a representation of the experimental apparatus to help resolve some of the arguments.

Clamp the test tube at the top of the test tube with the tube at about a 30° angle.

Sentence-level:

Feasibility: moderate – simple instruction with clear argument roles. Constructing symbolic representations for the arguments may be tricky, particularly "the top of the test tube" and "with the tube at about a …" The second argument will likely be especially difficult to interpret.
Change of state in the experimental apparatus would be useful to have in the domain knowledge base for future interpretation.
It may be useful to have a representation of the experimental apparatus to help resolve the arguments (particularly to help interpret "top of the test tube")

Make sure all of the solid is at the bottom of the test tube.

Sentence-level:

Feasibility: moderate/weak – We need to paraphrase "make sure" since it's an idiomatic expression not found in the training corpora. The coreference resolution system will be unable to resolve "solid," but it should be able to resolve test tube. The rules that we define for extracting symbolic representations from the dependency structures should be enough to represent "bottom of the test tube", although domain knowledge would be useful.

This is a sanity check, it's not clear that we should add this to the domain ontology

Domain knowledge would be useful in interpreting "bottom of the test tube" by having an understanding of position within a container.

Step 9

Check the system for leaks by removing the pinch clamp.

Sentence-level:

Feasibility: moderate – predicate "check" is found in PropBank with the proper sense, so the arguments should be easy to label. Resolving them is more difficult. It is not clear what leaks should resolve to, since it's a concept rather than an entity. It is also not clear from this sentence alone what exactly to check for, since it's not clear what the effect of removing the pinch clamp would be without domain knowledge.

No domain knowledge to extract

Domain knowledge may be useful in interpreting how to check the leaks, although it's not clear to me how it could be applied.

Make sure the exit tube stays below the beaker water level.

Sentence-level:

Feasibility: moderate/weak – need to paraphrase "make sure" since it's an idiomatic expression. Secondary predicate "stays" has many senses, and it's not clear that the SRL system will be able to find its arguments ("exit tube," "below", and "beaker water level"). It will be impossible to resolve "exit tube" using automatic methods; domain knowledge of the apparatus must be used.

No domain knowledge to extract

Domain knowledge of the apparatus would be useful for resolving "exit tube"

If no water flows, continue; otherwise, find out where the leaks are.

Sentence-level:

Feasibility: moderate– two statements here. The first has two predicates, "flows" and "continue." It should be reasonably simple to construct a predicate/argument representation for "if no water flows", but how to interpret "continue" is unclear. In the second statement, "find out where the leaks are" is a rather vague instruction, but by analyzing the dependency structure it is likely we could construct a predicate such as DISCOVER(LOCATION(LEAK)). It is unclear what to do with such a predicate however.

No domain knowledge to extract

No domain knowledge needed to interpret

Document-level:

This instruction is part of a sequence for step 9. It is predicated on the previous two steps being interpreted. We should have symbolic representations for the previous two steps to interpret this one.

Step 10

With the pinch clamp off and the exit tube under the beaker water level, equalize the pressure inside the bottle to atmospheric pressure by raising or lowering the beaker until the water levels in the beaker and bottle are the same.

Sentence-level:

Feasibility: weak – very complex sentence structure. This sentence has an instruction, a constraint, and an explanation of how to perform the instruction. There are many predicates and arguments in this sentence, and the syntactic/dependency structure (assuming that the output of the parser we use is correct) will likely be too complex for any set of hand written rules to extract the semantic relations out of it.

Assuming that we were able to represent the sentence semantics symbolically, then there is much that we could add to the domain ontology, including that one can equalize the atmospheric pressure inside the containers by balancing the water levels, and that the pressure in the bottle and beaker would then be equal.

Having a representation of the apparatus may help resolve the instances to which “the pinch clamp,” “exit tube,” and “beaker” refer.

Once level, attach the pinch clamp to the tubing closest to the beaker.

Sentence-level:

Feasibility: moderate – sentence has reasonably simple structure, SRL system and semantic representation-building rules should be able to handle this. The coreference resolution system should be able to resolve “beaker” and “tubing”, although the systems will likely be unable to resolve “Once level”, since in this constraint, there is an ellipsis (i.e. “Once level” means “Once the water level in the beaker is equal to the water level in the bottle”).

Change of state in apparatus setup should be added to the domain ontology.

Having a representation of the apparatus in the domain ontology would help resolve the references to the parts of the apparatus.

Step 11

Without draining the water from the end of the exit tube, carefully replace the beaker with an empty 100 mL graduated cylinder while removing the pinch clamp.

Sentence-level:

Feasibility: weak – complex sentence structure. Single instruction with two constraints. Three predicates (“replace”, “draining”, and “removing”), with three sets of arguments. The SRL system may be able to identify the predicates (assuming the system can label multiple predicates), and it may be able to identify and label the arguments to all the predicates (unlikely), it is very unlikely that we will be able to automatically identify which of the predicates are constraints and how to represent the relation between the constraints and the instructions. The argument identification/labeling stage will be difficult because there are several hidden arguments (for instance, the argument denoting from what to remove the pinch clamp is missing). The coreference resolution system will likely be able to resolve most of the noun phrases in the sentence.

Change of state in apparatus setup should be added to the domain ontology.

Having a representation of the apparatus in the ontology may help resolve some of the references to the parts of the apparatus.

Document-level:

There is another constraint that applies to this instruction, namely that which is described in the next sentence.

If some water from the tube drains into the graduated cylinder that is fine as long as there isn't a continuous flow.

Sentence-level:

Feasibility: weak – complex sentence structure, arguments not in canonical form. It is unclear which is the dominant predicate for the sentence. The coreference resolution system should be able to resolve "tube" and "graduated cylinder" without much difficulty, but it is very unlikely that we would be able to automatically build the symbolic representation of the semantics.

We can extract the constraint that there shouldn't be continuous flow of water, assuming we have a symbolic representation.

Domain knowledge unlikely to be helpful in interpretation.

Document-level:

This is a constraint on the previous instruction

Make sure the exit tube is down inside the graduated cylinder.

Sentence-level:

Feasibility: moderate – simple instruction, but we would need to paraphrase "make sure" as "verify." There are two arguments that the SRL system should be able to identify. The structure of the sentence is simple enough that the rules we would have should be sufficient for constructing a symbolic representation from the syntactic/dependency structures over the sentences.

No domain knowledge to extract

May be useful to have a symbolic representation of the apparatus in the ontology.

Step 12

With a Bunsen burner, gently heat the test tube contents.

Sentence-level:

Feasibility: strong – simple sentential structure, although the instrument argument is not in canonical position. Single predicate (heat), and two arguments ("test tube contents," and "Bunsen burner"). The coreference resolution system should be able to resolve them.

We can extract that once this step is complete, the test tube will be at least warm.

No domain knowledge needed to interpret on a sentence-level

Document-level:

We can infer the consequences of this instruction by having the fact that heating an entity causes it to become warm in the domain ontology.

Once the solid begins to melt, the reaction will be quite vigorous, so control the heating by continuously moving the flame.

Sentence-level:

Feasibility: weak – a simple instruction with an explanation in non-canonical position. Single predicate in the instruction ("control") with two arguments: the theme "heating," and the predicative argument "by continuously moving the flame." The SRL system should be able to label the arguments to "control" with their semantic roles, but it is unlikely that it will be able to label the arguments to "moving." Also, it is unlikely that any rules that we would come up with would be able to relate the instruction to the explanation. The coreference resolution system would have difficulty resolving the NPs without using domain knowledge.

May be able to extract that as the solid begins to melt, the reaction will strengthen and become faster, assuming that we have a good symbolic representation.

Domain knowledge would be very useful to resolve "flame" (we would need to know that Bunsen burners heat objects using a flame) and to resolve "solid" (we would need to know that the content in the test tube is a solid, and that as it gets heated it undergoes a state change and becomes a liquid).

Do not apply heat near the clamp or stopper as they will melt.

Sentence-level:

Feasibility: moderate/strong – simple instruction with an explanation (in canonical position). Ideally, we could paraphrase "apply heat" as simply "heat," as it would likely make it easier for the SRL system to label the arguments. The syntactic/dependency structure should be simple enough to allow for the rules that we define to extract the symbolic representation.

This is a constraint on all the instructions involving the Bunsen burner, so it would be useful to add to the domain ontology.

A representation of the apparatus would help in resolving "clamp" and "stopper."

If the reaction is too vigorous, remove the flame for a few seconds.

Sentence-level:

Feasibility: strong – instruction predicated on an observation. Simple sentence structure, the SRL system should have no difficulty in labeling the arguments, and the syntactic/dependency structure is simple enough to allow for the rules to construct symbolic representations of the semantics.

This is a constraint on all instructions involving the Bunsen burner, so it would be useful to add to the domain ontology.

Knowing that the Bunsen burner is the source of the flame would help in resolving flame.

You should see the liquid level in the graduated cylinder increase as gas is produced and water is displaced.

Sentence-level:

Feasibility: moderate/weak – an observational statement. It has a reasonably complex syntactic/dependency structure, with several predicates. The salient details we wish to encode are that the liquid level in the cylinder will increase because the gas produced pushes the water level up due to the effects of air pressure. It is unlikely that the SRL system will be able to label the arguments, because all of the passive dependent clauses in the sentence make it unclear which are the predicates. We may be able to cope with this through the rules we define to convert the syntactic/dependency structure to the semantic representations. The coreference resolution system should be able to resolve the references.

This would be very useful to add to the domain ontology, since it explains the effects of changing air pressure on the water level.

It would be useful to resolve “gas” to O_2

Once the bubbling has subsided, apply high heat to the test tube for 8-10 minutes to ensure the decomposition is complete and no more gas is produced.

Sentence-level:

Feasibility: moderate/weak – instruction with a prerequisite observation and an explanation. It has a complex syntactic/dependency structure that may be difficult to convert to its semantic representation. The passive statements may make it difficult to identify the predicates and to label their arguments. This has the same barriers to interpretation as the last instruction. The coreference resolution system should be able to resolve “test tube,” but it will be unable to resolve the other noun phrases.

Can extract that that the heat applied to the test tube causes the solid to decompose and release oxygen gas

Domain knowledge would be useful in resolving “gas” to O_2

Note that O_2 is still produced at a slower rate even when the bubbling has stopped as evidenced by the water level rising.

Step 13

Allow the test tube to cool leaving the apparatus alone (no pinch clamp, tubing left inside the graduated cylinder under the water level).

Sentence-level:

Feasibility: moderate/weak – There are actually two statements here. The first is the instruction to allow the test tube to cool, along with an explanation of how to do so. This is reasonably straightforward, although it’s not clear whether the SRL system would be able to label the arguments correctly, given that there is a missing word in the PP “leaving the apparatus alone.” The second statement “(no pinch clamp, tubing left inside the graduated cylinder under the water level)” will likely be impossible to interpret automatically due to the irregular sentence structures within. In fact, there are no identifiable predicates, making it impossible to label the arguments.

No domain knowledge to extract
No domain knowledge needed for interpretation

This should require 8-10 minutes to completely reach room temperature.

Step 14

Obtain the barometric pressure in the room.

Sentence-level:

Feasibility: strong – simple instruction, simple syntactic/dependency structure, single predicate with a hidden argument, a theme, and a location.
No domain knowledge to extract
It would be useful to know how to obtain the barometric pressure (i.e. by using a barometer).

Step 15

When the test tube is cool, slowly raise or lower the graduated cylinder (keeping the tubing under water) until the water levels in the cylinder and bottle are equal.

Sentence-level:

Feasibility: moderate/weak – instruction with two constraints and a stopping condition. The dominating predicate is a disjunction of two predicates ("raise" and "lower"), which the SRL system may have difficulty identifying. One of the arguments is predicative ("until the water levels in the cylinder and bottle are equal"), and we would need to apply rules to extract a semantic representation from the syntactic/dependency structure covering the argument.
No domain knowledge to extract
An understanding of the apparatus would be useful in resolving references to parts of the apparatus.

If needed, you may have to raise the bottle instead by carefully lifting the ring stand assembly.

Sentence-level:

Feasibility: moderate – instruction with explanation. The instruction's predicate, "raise," should be easy enough to identify with an SRL system, and it should be able to label the predicate's arguments. To construct a symbolic representation of the predicative argument "by carefully lifting the ring stand assembly," we would likely need to construct a symbolic representation by applying the set of rules to its syntactic/dependency structure. The coreference resolution system will likely have difficulty resolving "assembly."
May be able to extract knowledge that one can lift either the graduated cylinder or the bottle to equalize the water levels
It would be useful to have an understanding of the apparatus to resolve the references of "ring stand assembly" and "bottle."

Once the levels are equal, clamp the tubing with the pinch clamp.

Sentence-level:

Feasibility: strong – simple instruction with a prerequisite. Only hidden argument is the agent, and it is clear that "clamp" is the predicate. The SRL system should have no difficulty in labeling the arguments, the coreference system will likely be able to resolve "pinch clamp," "tubing," and possibly "levels." Furthermore, the syntactic/dependency structure is likely simple enough that we can get a good symbolic representation from applying the rules for extracting semantics from the structure.

This describes a change of state in the apparatus which may be necessary to interpret future instructions.

A representation of the apparatus in the domain may be helpful in resolving the references of "tubing" and "pinch clamp," and knowing that "levels" refers to the water levels in the bottle and test tube.

The mixture of gases in the bottle and test tube are now at the room's atmospheric pressure.

Sentence-level:

Feasibility: moderate/strong – simple informative statement with no hidden arguments. The predicate is the identity predicate ("are"), so labeling the semantic roles may be difficult. The dependency/syntactic structure of this is simple, so it should not be too difficult to automatically extract the symbolic representation from the parses. The coreference resolution system should be able to resolve "bottle" and "test tube" but may have trouble with "mixture of gases."

We could the fact that as a result of equalizing the water levels and closing the system (i.e. the previous two instructions), the pressure is equalized as well.

Domain knowledge not needed for interpretation.

Step 16

Carefully remove the tubing from the graduated cylinder so that the water inside the tube does not drain out.

Sentence-level:

Feasibility: moderate/strong – instruction with constraint. The syntactic/dependency structure is straightforward enough that we should be able to identify the predicate "remove" and correctly label the arguments ("tubing", "graduated cylinder"). It is not clear how to construct a symbolic representation of the constraint, particularly, how to represent "drain out." The syntactic structure is simple enough, however, that with sufficient rules we may be able to construct the symbolic representation of the constraint. The coreference resolution system should be able to resolve the references of "graduated cylinder," "tubing," and perhaps "water inside the tube," although having a representation of the apparatus in the domain ontology would likely be of great help.

Change of state in apparatus would be necessary to interpret future instructions

Knowledge of apparatus may be useful in resolving references.

Determine the water level in the graduated cylinder, reporting the value with the correct number of significant figures.

Sentence-level:

Feasibility: moderate– Two instructions. The syntactic structure is relatively simple, and both predicates "determine" and "reporting" are in the training corpus, so we will likely be able to label the arguments to "determine" correctly (although it's not clear that we will be able to label the arguments to the second predicate accurately). The syntactic/dependency structure is reasonably straightforward, so we may be able to automatically convert the structure into semantics. We should also be able to resolve the references of "graduated cylinder" and "water level" without too much difficulty.

No domain knowledge to extract

We need to be able to know how to determine the water level, namely by measuring the height from the water level to the bottom of the cylinder. This should be in the domain ontology. We should also have an understanding of significant digits, and measuring properties to an acceptable standard.

Measure the temperature of the water.

Sentence-level:

Feasibility: strong – Simple instruction, simple syntactic structure. Coreference resolution system should be able to resolve "water" from the immediate context.

Could add the temperature of the water to the domain knowledge bank

Would be useful to have a predicate saying how to measure temperature (i.e. with a thermometer) in the domain ontology.

Document-level:

Execution of this instruction must follow the step 15 and all of its prerequisite steps. We need an up-to-date representation of the experimental state.

At this temperature, obtain the vapor pressure of water from a reference.

Step 17

Carefully remove the test tube from the clamp and weigh it on the analytical balance.

Sentence-level:

Feasibility: moderate – there are two instructions here, to first remove the test tube from the clamp and then to weigh it. There are two predicate/argument structures in the sentence, which the SRL system may be able to label (it should be able to label at least one of the predicate/argument structures). The syntactic/dependency structures should be simple enough that our rules would be able to construct semantic representations. The difficulty would be for the coreference resolution system to resolve "it."

No domain knowledge to extract.

It would be useful to use the knowledge extracted in step 5 to know how to weigh the test tube on the analytical balance.

Bibliography

Alfonseca, E., & Manandhar, S. (2002). An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery. *Proc. International Conference on General WordNet* .

Bikel, D. (2004). Intricacies of Collins' Parsing Model. *Computational Linguistics , 30* (4), 479-511.

Collins, M. (1996). A new statistical parser based on bigram lexical dependencies. *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (pp. 184-191). Santa Cruz, CA: Association for Computational Linguistics.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing.* PhD Thesis, University of Pennsylvania.

Collins, M. (1997). Three Generative, Lexicalised Models for Statistical Parsing. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (pp. 16-23). Madrid, Span: Association for Computational Linguistics.

Collins, M., & Koo, T. (2005). Discriminative Reranking for Natural Language Parsing. *Computational Linguistics , 31* (1), 25-69.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., et al. (1005). Unsupervised Named Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence , 165* (1), 91-134.

Evans, R. (2003). A Framework for Named Entity Recognition in the Open Domain. *Proc. Recent Advances in Natural Language Processing* .

Fillmore, C. J., Johnson, C. R., & Petruck, M. R. (2002). Background to FrameNet. *International Journal of Lexicography , 16*, 235-250.

Furstenau, H., & Lapata, M. (2009). Semi-Supervised Semantic Role Labeling. *Proceedings of the 12th Conference of the European Chapter of the ACL* (pp. 203-228). Athens, Greece: Association for Computational Linguistics.

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics , 28* (3), 245-288.

Haghighi, A., & Klein, D. (2007). Unsupervised Coreference Resolution in a Nonparametric Bayesian Model. *Proceedings of the 45th Annual Meeting of the*

Association of Computational Linguistics (pp. 848--855). Prague, Czech Republic: Association for Computational Linguistics.

Huang, L. (2008). Forest Reranking: Discriminative Parsing with Non-Local Features. *Proceedings of ACL-08: HLT* (pp. 586--594). Columbus, Ohio: Association for Computational Linguistics.

Jircitano, A. (n.d.). *The Decomposition of Potassium Chlorate*. From http://chemistry.bd.psu.edu/jircitano/KClO3decomp05.pdf

Kipper-Schuler, K. (2005). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon.* PhD Thesis, University of Pennsylvania.

Klein, D., & Manning, C. D. (2003). Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics* (pp. 423-430). Sapporo, Japan: Association for Computational Linguistics.

Lomax, J., Dillner, D., & Streib, D. (n.d.). *Experiment 9B: Studying the Decomposition of Potassium Chlorate*. From Midshipman Online Lab System: http://www.usna.edu/ChemDept/plebeChem/manual/Ex9B%20KClO3%20and%20gas%20law.pdf

McDonald, R., Pereira, F., Ribarov, K., & Hajic, J. (2005). Non-Projective Dependency Parsing using Spanning Tree Algorithms. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (pp. 523--530). Vancouver, British Columbia, Canada: Association for Computational Linguistics.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes , 30* (1), 2-26.

Narayanaswamy, M., Ravikumar, K., & Vijay-Shanker, K. (2003). A Biological Named Entity Recognizer. *Proc. Pacific Symposium on Biocomputing , 8*, 415-426.

Ng, V. (2007). Shallow semantics for coreference resolution. *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, (pp. 1689--1694). Hyderabad, India}.

Ng, V. (2008). Unsupervised Models for Coreference Resolution. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 640--649). Honolulu, Hawaii: Association for Computational Linguistics.

Nivre, J. (2007). Incremental Non-Projective Dependency Parsing. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference* (pp. 396-403). Rochester, New York: Association for Computational Linguistics.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics , 31* (1), 71-106.

Ponzetto, S. P., & Strube, M. (2006). Semantic Role Labeling for Coreference resolution. *11th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 143-146). Association for Computational Linguistics.

Soon, W. M., Ng, H. T., & Lim, D. C. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics , 27* (4), 521-544.

Taskar, B., Klein, D., Collins, M., Koller, D., & Manning, C. (2004). Max-Margin Parsing. *Proceedings of EMNLP 2004* (pp. 1-8). Barcelona, Spain: Association for Computational Linguistics.

Toutanova, K., Haghighi, A., & Manning, C. D. (2008). A Global Joint Model for Semantic Role Labeling. *Computational Linguistics , 34* (2), 161-191.

Turian, J., Wellington, B., & Melamed, I. D. (2006). Scalable Discriminative Learning for Natural Language Parsing and Translation. *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS).* Vancouver, Canada.

Versley, Y., Ponzetto, S. P., Poesio, M., Eidelman, V., Jern, A., Smith, J., et al. (2008). BART: A Modular Toolkit for Coreference Resolution. *Proceedings of the ACL-08: HLT Demo Session* (pp. 9-12). Columbus, Ohio: Association for Computational Linguistics.