Step Semantics: Representations for State Changes in Natural Language

Ken Forbus[1], Maria Chang[2], Danilo Ribeiro[1],

Tom Hinrichs[1], Maxwell Crouse[1], Michael Witbrock[2]

[1]Qualitative Reasoning Group, Northwestern University, 2233 Tech Drive, Evanston, IL, 60208
[2]IBM TJ Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, NY 10598
forbus@northwestern.edu

Abstract

The dynamics of the world is often bound up in processes. These include continuous processes, such as flows and motion, and discrete processes, such as count and break. Things that occur in the world can often be described at multiple levels of detail, using combinations of continuous and discrete processes, and it is important to be able to shift among levels of detail as needed for communication and understanding. This paper describes *step semantics,* a framework that draws upon prior work in qualitative reasoning and discrete action representations to provide a set of representation conventions for processes described in natural language, independent of a particular task or dataset. We explore its potential in two ways: Analyses of recipes with complex temporal structure and learning from AI2's ProPara dataset.

Introduction

Human level complex question answering requires deep understanding of processes and procedures. These processes can include continuous quantities, like speed, or discrete quantities, like integer counts. Moreover, processes and their sub events are often described at different levels of detail. For example, "cook dinner" can be viewed as a discrete event, but it can involve many instances of continuous processes (e.g. mixing, splitting, heating, cooling) when viewed at a finer level of detail. Similarly, the life cycle of a frog might be described in terms of three discrete states: eggs, tadpoles, and adults, even though the growth of legs in a tadpole and the shrinkage of its tail happen smoothly over many days. Question-answering systems need to be able to represent both discrete and continuous processes and reason about them in ways that are compatible with each other.

Although considerable advances have been made in reasoning for question-answering, understanding processes is still a major challenge. Few datasets include questions that require inference about processes, and most are in the domain of science tests, e.g. ARC (Clark et al. 2018) & ProPara (Dalvi et al. 2018). These datasets are steps in the right direction, but there are more subtle phenomena that they do not test, as explained below.

This paper presents a representation for processes described in text that combines qualitative process theory (Forbus, 1984) with models of discrete actions and change from OpenCyc and FrameNet to go beyond what either could do alone. We show the utility of this synthesis by examining both recipes, which can incorporate complex temporal structure and combinations of continuous processes and discrete actions, and learning from AI2's ProPara dataset. We argue that this synthesis provides a prerequisite for human-level reasoning for answering questions about processes.

Background & Related Work

To provide a set of representation conventions for processes described in natural language, we draw upon prior work in qualitative reasoning and discrete action representations, which we summarize here.

Qualitative Process Theory

Qualitative process theory is a representational system for describing continuous processes. Processes provide a notion of mechanism, in that, aside from the actions taken by agents, ultimately all changes are explainable in terms of the effects of processes. This strong inductive bias simplifies learning and conceptual change (e.g. Friedman et al. 2017). Liquid flow, for example, happens between a source and destination. Its direct effects – direct influences – are specified as part of the process. For example, in liquid flow,

```
(I+ (AmountOf ?dest) (FlowRate ?lf))
(I- (AmountOf ?src) (FlowRate ?lf))
```

That is, the amount of liquid in the source is decreased by the flow rate of the liquid flow, and the amount of liquid in the destination is increased by the same rate. Processes are *active* when their conditions are satisfied, e.g. for liquid flow, when the pressure in the source is greater than the pressure in the destination. Continuous processes are typically expressed in language via verbs, e.g. flow, move. Participants are typically described in language via role information about the verb, including prepositional phrases. For example, "Water flowed out of the bathtub onto the floor." describes a liquid flow whose source is the bathtub and whose destination is the floor. Notice that the path is implicit: This is a common property of language, we tend to leave implicit things that are not important or are inferable by the listener.

Causal laws associated with objects support inferring the indirect effects of processes via *qualitative proportionalities*, which are partial information about functional dependencies. For example, in a contained liquid (Hayes, 1984),

```
(qprop+ (Level ?l) (AmountOf ?l)
(qprop+ (Pressure ?l) (Level ?l))
```

That is, a change in amount will cause a change in level, which in turn will cause a change in pressure.

Quantities in QP theory are described in terms of ordinal relationships with other quantities, where the relevant set of comparisons is automatically derived from the structure of the domain theory. For example, a liquid which has a fluid path to other liquids will lead to their relative pressures being tracked, because that is part of what determines if liquid flow is active. If phase changes such as freezing and boiling are under consideration, the temperature of the liquid will also be compared with its melting and boiling points. These *limit points* are often mentioned in texts, e.g. "When all the water is drained from the pasta…" While sometimes specific numerical values are known (e.g. "Cook the roast until its internal temperature is 165 degrees."), often they are not (e.g. "Wait until the mixture has cooled.").

Qualitative representations carve time up into discrete units, based on when qualitative properties change. Following Hayes (1984), we represent changes over time in terms of *histories*, which are pieces of space-time over which the qualitative properties of some set of objects is the same. For instance, the cooking episode in the history of the creation of a roast starts when the roast is placed in the oven and ends when it is removed. Its spatial aspect is the union of the spatial aspects of the participants in it, e.g. the oven and the roast. We note that, in many qualitative reasoning projects, a more global notion of qualitative state is often used, where all of the entities under consideration are lumped together. We prefer using histories here because they allow for finer-grained decomposition of behavior that seems more suitable to the level of partial information found in language.

Events and Discrete Actions

To represent events, we draw upon a combination of concepts from FrameNet (Ruppenhofer et al. 2016) and the OpenCyc ontology. Specifically, we use neo-Davisonian representations, where events are reified and role relations are used to describe their particular aspects, such as participants, location, and duration. For example, consider the word "convert". In Cyc conventions, the word itself is denoted by an entity (i.e. `Convert-TheWord`). FrameNet has four senses of convert when used as a verb, which draw on three semantic frames (i.e. `FN_Undergo_transformation`, `FN_Cause_change`, and `FN_Exchange_currency`). Each sense is also linked in the KB to an event from the Cyc ontology (i.e. `Converting-Something`, `Convincing-CommunicationAct`, `CurrencyExchange`, `IntrinsicStateChange`). The FrameNet information provides two valuable sources of information for supporting natural language understanding. The first is a mapping from lexemes (i.e. word senses) to frames. For example, eight lexemes evoke the `FN_Creating` frame. The second are a set of *valence patterns* that help constrain parsing by stating what patterns of auxiliary phrases are common. The OpenCyc information provides semantic constraints, including type information, allowable role relations, and inference rules concerning that type of event.

We assume events take time, although for some perspectives, that time is so short that it can safely be treated as an instant (Allen & Hayes, 1990). Events whose internals are irrelevant to understand a particular text can be considered as discrete actions. To provide the inferential semantics for discrete actions, we assume STRIPS operators (Fikes & Nilsson, 1971) for simplicity.

We note that in the Qualitative Reasoning community, there have been several prior efforts that integrate discrete and continuous models of actions and processes, albeit for very different purposes. Hogge (1987) described how QP descriptions of processes could be compiled into operators for use with a temporal planner. Forbus (1989) explored how STRIPS operators could be added to envisionments based on QP theory, to simulate systems that incorporated actions alongside physical processes. Drabble (1993) showed how QP theory could be combined with an HTN planner to both generate and execute plans involving both actions and processes. None of these prior efforts addressed integrating continuous and discrete representations in understanding natural language, which is our focus here.

Answering Questions about Processes

Reading comprehension is largely evaluated through question answering tasks. State of the art performance on these tasks is generally achieved using artificial neural networks that take a query and context (e.g. a paragraph) as inputs and predict a span of text within the context that contains the answer (e.g. Chen et al. 2017, Seo et al. 2017). However, by definition this poses a challenge when the answer to a question is not explicitly stated in the source context paragraph. In other words, questions that require inference to ascertain implicit information are still a challenge. This is illustrated by several new datasets that require more sophisticated reasoning, like tracking state changes in processes (ProPara), and a host of other knowledge and reasoning types (ARC). An analysis on a subset of the ARC dataset suggests that a large proportion of questions (99/192, 52%) involve causal or physical knowledge (Boratko et al. 2018). An analysis by Crouse & Forbus (2016) suggests that 29% of the problems in 4th grade science tests require qualitative reasoning of the form QP theory provides.

The ProPara dataset (Dalvi et al. 2018) is the first large dataset of human generated natural language paragraphs about processes that are annotated with status, step, and location of participating entities. Along with the ProPara dataset, Dalvi et al. (2018) introduced two artificial neural network models to track state changes: a system that uses bilinear attention over sentences and an end-to-end system that uses bilinear attention over the entire paragraph. As of this writing, the two most successful models for ProPara enhance neural reading approaches with rules or knowledge graphs. Tandon et al. 2018 characterized ProPara as a structured prediction task, using commonsense rules derived VerbNet to avoid unlikely answers. Das et al. 2018 achieved state of the art results by recurrently building dynamic knowledge graphs that track entity locations. Das et al. 2018 also evaluated their system on a dataset of natural language recipes (Kiddon et al. 2018), which had previously been interpreted with neural process networks that simulate recipe actions and their effects (Bosselut et al. 2018). These recent papers suggest that commonsense knowledge and structured representations (e.g. in the form of knowledge graphs in Das et al. 2018 or domain-specific state predictors in Bosselut et al. 2018) are important for understanding the many complex aspects of procedural texts. We use ProPara to explore the step semantics framework and to understand how it can support some of these additional aspects of process understanding.

Step Semantics

Language is a blunt instrument. The challenge of learning by reading is to assemble, from both the signal in texts and the reader's preexisting knowledge, a reasonable extension of that reader's knowledge. Step semantics is a framework for specifying what a reader should learn from the language describing the steps of a process. Importantly, language enables people to intermingle continuous and discrete descriptions, hence our drawing together continuous processes, discrete actions, and events to provide the representational capacity necessary.

We call our account step semantics for two reasons. First, it is about the steps in a process viewed as a sequence of operations or events. (Operations, for recipes and procedures, events for natural processes that can be decomposed, such as life cycles and the formation of rain.) Second, often the internal structure of a step relies on one or more continuous processes, i.e. representable via the notion of process in qualitative process theory. At a coarser grain of description, the continuous changes are summarized via step changes (Rickel & Porter, 1994).

Ontology

We assume that a natural language description of a process consists of a sequence of sentences. The understanding process must create a description of states and steps. By state we mean an episode in a history (Hayes 1984), i.e. a set of propositional statements, including fluents, that is taken to hold over some time (instant or interval) describing a set of individuals. By step, we mean an event, or a set of events, that describes what happens during the transition between its before state and after state. The before/after relations impose a sequential ordering on states. This ordering can be cyclic, as in oscillations or life cycles. There can be alternate steps from a state, corresponding to events that either are alternatives to each other (e.g. bake in a microwave versus bake in an oven) or are occurring in parallel (e.g. the sprouting of legs and shrinking of its tale occurring at the same time in a tadpole's maturing).

The relationship between sentences and steps can be complicated. In the simplest case, e.g. ProPara, each sentence is assumed to be a single step, each state has at most one step leading to it and at most one step leading from it, and the order of events is given by the order of sentences. None of these assumptions hold more generally. The mapping between sentences and steps can be one to many. In the other direction, a step can be spread across multiple sentences in language. The incremental nature of natural language is why learning by reading systems using QP theory rely on a frame-based equivalent notation (McFate et al. 2014). In complex processes, e.g. recipes, steps can be undertaken in parallel (e.g. creating gravy while roasting a turkey), and can include multiple next steps (e.g. the reason to separate eggs is to do something different with the yolks versus the whites), and multiple previous steps (e.g. combining parts created by earlier steps). The temporal order in the events being different from the sequence of sentences describing them is very common in fiction, but is also used in instruction as a motivation. For instance, stories about why Hawaii caught a lucky break when Hurricane Lane dropped from a category 5 to a category 2 storm typically started with the good news and then described why this was such good news.

It should be clear from these complexities that understanding processes expressed in text, despite whatever progress is made on ProPara, remains a challenging problem.

Features

There are four fundamental kinds of steps:

- Changes of existence: A step can create or destroy something.
- Changes of property: A move step changes the location of something, for instance, and painting changes its color. Transformations, e.g. phase transformations such as boiling, change the type of an object.
- Change of quantity: A quantity change step indicates that the given quantity has risen or fallen during the step. The continuous processes that are causing this are often implicit. This is a useful thing to say if there are competing continuous processes occurring during a step, since knowing the result on a parameter of interest provides information about the relative magnitudes of effect. For example, evaporation from a bathtub is swamped by the change in mass from even a small stream of water flowing into it.
- Occurrence of a sub-process: A subprocess step describes the changes wrought by some process occurring within the larger process being described. For instance, if the water cycle is the process being described, there will typically be steps describing the roles of evaporation, condensation, and precipitation as part of that description.

These four types are mutually exclusive. As noted above, a single sentence may imply multiple steps, and a single step might be communicated by multiple sentences. A system with broad knowledge of the world will have representations encompassing multiple levels of detail and incorporating multiple perspectives (Falkenhainer & Forbus, 1991). This vocabulary of steps provides an interface layer between language and these representations, the specific level of detail and perspective depend on the level of detail in the natural language description. For instance, consider a moving object that is part of a larger mechanical system. Its movement might be simply described as a single change in property (i.e. location) step, or it may be described as a sub-step in the larger, more detailed description of the entire system.

Inertia is assumed for existence and property changes, i.e. if something exists then it continues to do so, until explicitly terminated or changed by some other step. Quantity changes, on the other hand, are subject the operations of continuous processes – one cannot melt chocolate, for example, and then leave it on the counter for an hour and assume it will remain molten.

Part of the hierarchy of process descriptions arises from hierarchies in place descriptions. In describing photosynthesis, for example, chloroplasts might be described as "in the leaf". A common heuristic is that the location of a process has to include the location of all of the constituents being used in it. So, the creation of sugar happens in the leaves, while the process as a whole must also include the roots and stems, since they collect and transport water that are used in the process.

When fluids are involved, we have found that both the classic piece of stuff and contained fluid ontologies (Hayes, 1984) are useful. In linear (cyclic or acyclic) steps, the moving liquid can be characterized in terms of molecular collections (Collins & Forbus, 1987), i.e. a specialization of the piece of stuff ontology such that the fluid moving is considered to be large enough to have macroscopic properties (e.g. temperature and pressure in moving water or air), but so small as to maintain coherence (e.g. not split at a fork in a piping system).

Connection to Language

We use FrameNet as a bridge between continuous processes and language (McFate & Forbus, 2016).

We note that there are many complexities in carving up constituent processes in language. For example, "Roots absorb water and minerals from the soil." Should this be viewed as two separate absorption processes, one for water and one for minerals? Without either additional knowledge or additional explanation, it is impossible to tell. Liquids are often used for transporting other things, in suspension or solution, in biological and engineered systems, and if the next sentence continues with "This combination of water and mineral flows…", then this expectation is satisfied. But in general there will be multiple possible interpretations which need to be maintained (or regenerated on backtracking) to understand such explanations. We begin by examining how simple steps can be recognized in terms of the verbs used in a sentence, then discuss how the semantics of verbs linked to processes can be used to extract additional information about a step.

Creation Steps: These are represented by the FrameNet frame `FN_Creating` and the linked Cyc event type `CreationEvent`. For biological creatures, the corresponding linked frames are `FN_Giving_birth` and `BirthEvent`. We note that FrameNet does not treat giving birth as a subframe of creating, but since Cyc does include `BirthEvent` as a specialization of `CreationEvent`, we treat this as a subcategory. The lexemes for this frame include create, assemble, form, formation, generate, make, produce, and several others.

Destruction Steps: These are represented by the FrameNet frame `FN_Destroying`, and the linked Cyc event type `DestructionEvent`. There are subframes for biological creatures, e.g. `FN_Killing`, `KillingByOrganism`.

Property Change Steps: There are quite a variety of these, e.g. `FN_Cause_change`, which can apply to names, religious beliefs, political climates, and so on. Similarly, `FN_Change_of_phase_scenario` covers phase changes such as freezing, boiling, and solidifying.

Quantity Change Steps: These include frames such as `FN_Change_of_temperature`, which covers verbs such as heat, warm, cool, chill, and refrigerate, and `FN_Change_position_on_a_scale`, which covers verbs such as rise, balloon, fluctuate, increase, etc.

Subprocess/Event Steps: Examples include `FN_Motion`, `FN_Fluidic_motion`, and `FN_Giving`. The role relations describe changes in the participants, e.g. prepositional phrases involving "from" and "to" identify the source and destination of something whose physical location or ownership changes.

Part of the complexity of natural language understanding of process descriptions comes from unpacking steps from the semantic interpretation. Another source of complexity is assembling a set of plausible temporal relationships among the steps. ProPara attempts to simplify these issues by treating each sentence as representing a single step, and assuming a strict identification of order of sentences with order of events that they describe. (An exception consists of cycles, where language like "continuing the cycle" indicates the existence of a cycle, but this lies outside the semantic representations stipulated in ProPara.)

Examples

To illustrate how step semantics can be used for natural language understanding, we use examples from the domains of recipes and ProPara.

Recipes

Consider the following recipe for French toast[1]:

1. In a small bowl, combine, cinnamon, nutmeg, and sugar and set aside briefly.
2. In a 10-inch or 12-inch skillet, melt butter over medium heat. Whisk together cinnamon mixture, eggs, milk, and vanilla and pour into a shallow container such as a pie plate. Dip bread in egg mixture. Fry slices until golden brown, then flip to cook the other side. Serve with syrup.

Despite being short and simple, this recipe includes several different types of steps (including some that are not explicitly stated) and is written in such a way that we cannot rely on steps being properly enumerated or being separated into distinct sentences. In the first step, the three dry ingredients (cinnamon, nutmeg, and sugar) undergo a mixing process which makes them individually irrecoverable. The second numbered step includes several actions (melting, whisking, pouring, dipping, frying, and flipping). While melting, butter undergoes changes in property (i.e. phase) and quantity (i.e. temperature). In whisking, individual wet and dry ingredients are combined and (in the same sentence) are poured into another container. In the bread-dipping step, ingredients are not destroyed, but there are changes in property (i.e. moisture and location). In the following sentences, multiple steps (frying, flipping) are combined into one sentence. In frying and flipping, there is a qualitative limit point ("until golden brown") and there is an implicit change of location before the final serving step.

Lexeme	*FrameNet Frames*	*Entities involved*

Combine	`FN_Amalgamation,`
	`FN_Creation`
		Cinnamon, nutmeg, sugar

Melt	`FN_Change_of_temperature,`
	`FN_Change_of_phase_scenario`
		Butter

Whisk	`FN_Self_motion,`
	`FN_Amalgamation,`
	`FN_Creation`
		Cinnamon mixture, eggs, milk, vanilla

Dip	`FN_Dunking`	Bread, egg mixture

Fry	`FN_Apply_Heat,`
	`FN_Change_of_temperature,`
	`FN_Amalgamation`
		Slices, (melted) butter

Flip	`FN_Move_in_place`	Bread

Table 1: Lexemes, frames, and entities for French toast recipe.

This recipe also exhibits a subtle temporal structure. Melting the butter does not need to happen before mixing ingredients and dipping bread, but all of those things need to happen before frying. These constraints cannot be inferred by the order that each of the steps is introduced, since each step does not necessarily depend on all steps previously mentioned. Instead, they can be inferred by reasoning about the entities involved in each step and their properties. Table 1 shows how individual lexemes and frames can be used to characterize each step.

The following recipe for roasted brussels sprouts[2] illustrates another complex temporal structure:

1. Preheat oven to 400 degrees F.
2. Cut off the brown ends of the Brussels sprouts and pull off any yellow outer leaves. Mix them in a bowl with the olive oil, salt and pepper. Pour them

[1] https://www.foodnetwork.com/recipes/robert-irvine/french-toast-recipe-1951408

[2] https://www.foodnetwork.com/recipes/ina-garten/roasted-brussels-sprouts-recipe2-1941953

> on a sheet pan and roast for 35 to 40 minutes, until crisp on the outside and tender on the inside. Shake the pan from time to time to brown the sprouts evenly. Sprinkle with more kosher salt (I like these salty like French fries), and serve immediately.

Unlike the French toast recipe, this recipe describes steps such that each step necessarily begins before steps that are described later. However, the roasting step is supposed to temporally subsume the pan-shaking step (even though the recipe lacks a phrase like "while the sprouts roast…" to explicitly indicate that one step occurs during another). One way to make this inference is to identify the goal of pan shaking as a color property change of the sprouts (i.e. "browning") that is the ending condition for the roasting step.

These recipes are both relatively short and straightforward. However, they illustrate that (1) steps are not necessarily executed in the order that they are described, (2) that steps that are described with a single lexeme can denote multiple types of change (e.g. temperature and phase), and (3) that understanding the temporal constraints between steps hinges on the semantics of the processes (e.g. roasting, browning) and entities involved (e.g. pan).

ProPara

ProPara consists of 488 paragraphs about processes and a set of parameterized questions about the participants in each process paragraph. These questions concern when an entity is created, destroyed, or moved. Consider the following paragraph from the ProPara dataset:

"Chloroplasts in the leaf of the plant traps light from the sun. The roots absorb water and minerals from the soil. This combination of water and minerals flows from the stem into the leaf. Carbon dioxide enters the leaf. Light, water and minerals, and the carbon dioxide all mix together. This mixture forms sugar (glucose) which is what the plant eats. Oxygen goes out of the leaf through the stomata."

After reading this paragraph, a system ought to be able to answer questions like this one:

Q: Where is sugar produced?

A: In the leaf.

Our approach to answering these questions is to start with a general-purpose semantic parser, using a large knowledge base (NextKB[3]) and rich semantic interpretations based on Discourse Representation Theory (Kamp & Reyle, 2013), and use training data to customize the interpretation process for question answering. We call this *analogical Q/A training*. This approach has been used before on Geoquery (Crouse et al. 2018a), getting state of the art results with less data than typically required, and learning to recognize physical processes in paragraphs from science test questions (Crouse et al. 2018b). We combine this approach with step semantics to learn entailments from ProPara training data. The rest of this section describes how we do that and our preliminary results.

Analogical Q/A training works by taking natural language questions and some form of answers, and produces cases (i.e. sets of logical statements) that are retrieved and used in subsequent question answering. Typically natural language answers are provided, but here we use the table format provided by AI2, translated into predicate calculus, as shown in Figure 1. In training, the system is learning to map the FrameNet/Opencyc semantics it constructs to instances of events from the categories CreationEvent, DestructionEvent, and MovementEvent. We call it analogical Q/A training because what is created during the learning process are *query cases*, which are simple cases that provide a bridge between the logical forms produced by language and representations about processes. Queries to answer questions are generated by applying and composing query cases via analogy to interpret new texts.

(isa participant123 Participant)
(isa event123 CreationEvent)
(outputsCreated event123 participant123)
(outputsCreatedLocation event123 tolocation123)

(isa participant123 Participant)
(isa event123 DestructionEvent)
(inputsDestroyed event123 participant123)

(isa participant123 ProParaParticipant)
(isa fromlocation123 Location)
(isa tolocation123 Location)
(isa event123 MovementEvent)
(objectMoving event123 participant123)
(fromLocation event123 fromlocation123)
(toLocation event123 tolocation123)

Figure 1. Target logical form for each possible state change.

The process of constructing query cases during training works like this: First, the NLU system generates a set of syntactic and semantic choices, representing the space of possible interpretations for each sentence. Second, mappings between this space of interpretations and the target semantics (i.e. one of the three choices in Figure 1) are constructed. This involves using structural relations in the KB to find paths between concepts and relations. For example, here is a path that indicates that pulling is a kind of motion:

[3] NextKB integrates the OpenCyc ontology with FrameNet contents, a large lexicon, and support for qualitative and analogical reasoning. It will be available as a creative commons attribution resource shortly.

PullingAnObject → CumulativeEventType →
Movement-Rotation → MovementEvent.

Role relations from the semantic interpretation are mapped to roles in the target logical form by using inheritance relations involving predicates, e.g.

objectActedOn → EventOrRoleConcept →
objectMoving.

Typically there will be multiple potential matches, and these are filtered and scored based on constraints from Gentner's (1983) structure-mapping theory (e.g.1:1 mappings, prefer more systematic structures), albeit with re-representation occurring as part of the processing, similar to (Fan et al. 2009). The final step constructs query cases from these connections, and stores them into a case library for subsequent retrieval during Q/A.

Question-answering during testing proceeds as follows. Each test paragraph is read sentence by sentence. For each sentence, for each participant *p*, the following three categories of queries are asked: (Cat-1) Is *p* created (destroyed, moved) in the process? (Cat-2) When is *p* created (destroyed, moved)? (Cat-3) Where is *p* created (destroyed, moved from/to). These queries are processed by using analogical retrieval from the case library constructed during training. The best query cases are instantiated and ranked according to how well they match the sentence semantics. The consequents of the highest ranked query cases are used to predict the state change of the queried participant. Finally, all state changes are aggregated and the following common sense rules are applied to propagate the states of each participant:

1. Inertia: states are propagated, both forward and backwards, until a new state change occurs.
2. Collocation: If a participant X is converted to participant Y (X is destroyed when Y is created), and the position of Y is not known, then we assign the previous position of X to Y.

The combination of the queries and the common sense rules are used to generate a state change grid, in the format used by AI2, to compare against their answers. Table 2 compares our results on this task with the following models: ProComp (Clark et al. 2018), ProLocal, ProGlobal (both from Dalvi et al. 2018), ProStruct (Tandon et al. 2018), and KG-MRC (Das et al. 2018). Results are displayed as F1 scores for each category, as well as their respective macro-average. The ProStruct metric is different as the task was formulated as a structured prediction task.

While better than the prior rule-based model on two out of three categories, our approach does not yet out-perform the artificial neural network models, although it does better than ProLocal on two out of three categories, and better than all of them on Cat-2 questions. We believe there are at least two reasons for this. The first is the paucity of information extracted in cases currently, which does not provide enough discrimination during analogical retrieval, considerably reducing our recall score. We plan on exploiting more of the ontology and FrameNet information to address this. The second factor is that we were neither using coreference resolution nor the full set of commonsense rules used by the AI2 systems.

	Model	Cat-1	Cat-2	Cat-3	Macro averaged
Rule Based	ProComp	57.14	20.33	2.40	26.62
Artificial Neural Networks	ProLocal	62.65	30.50	10.35	34.50
	ProGlobal	62.95	36.39	35.90	45.08
	ProStruct	-	-	-	53.70*
	KG-MRC	62.86	40.00	38.23	47.03
Step Semantics	Our Model	49.50	43.92	17.13	36.85

Table 2. Comparison between models on ProPara dataset. Displayed values are F1 scores for each category, which are then macro-averaged. *ProStruct uses a different metric from previous papers.

Discussion

In this paper we propose a framework for representing state changes that occur in natural language descriptions of processes and procedures. Our analysis of recipes and learning experiment with process paragraphs suggests that this framework is capable of capturing some information from texts about processes.

We see several important lines of future work. First, we need to explore the ideas for improving ProPara performance noted above, to see how far we can push analogical Q/A training. It would not surprise us to find that exploiting additional linguistic and world knowledge during comprehension would lead to significant improvements. Second, we need to integrate step semantics into our learning by reading system, thereby enabling it to handle processes and procedures that go beyond ProPara, such as recipes including explicit cycles, forks, and joins, as well as moving beyond the 1:1 sentence/step model. These lines of work will, we hope, contribute to an account of human-level reasoning for question-answering about processes and procedures.

Acknowledgements

This research was supported in part by the Machine Learning, Reasoning, and Intelligence Program of the Office of Naval Research.

References

Allen, J.F. and Hayes, P.J (1990). Moments and Points in an Interval-Based Temporal Logic. Computational Intelligence, January.

Boratko, M., Padigela, H., Mikkilineni, D., Yuvraj, P., Das, R., McCallum, A., Chang, M., Fokoue-Nkoutche, A., Kapanipathi, P., Mattei, N., Musa, R., Talamadupula, K., and Witbrock, M. (2018). A systematic classification of knowledge, reasoning, and context within the ARC dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 60–70. Association for Computational Linguistics.

Bosselut, A., Levy, O., Holtzman, A., Ennis, C., Fox, D., & Choi, Y. (2018). Simulating action dynamics with neural process networks. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chang, M.D. and Forbus, K.D. (2015). Towards Interpretation Strategies for Multimodal Instructional Analogies. *Proceedings of the 28th International Workshop on Qualitative Reasoning* (QR2015). Minneapolis, MN.

Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to Answer Open-Domain Questions. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., Tafjord, O. (2018). Think you have solved Question Answering? Try ARC, the AI2 Reasoning Challenge. arXiv:1803.05457v1.

Clark, P., Dalvi, B., & Tandon, N. (2018). What Happened? Leveraging VerbNet to Predict the Effects of Actions in Procedural Text. arXiv preprint arXiv:1804.05435.

Collins, J. and Forbus, K. (1987). Reasoning about fluids via molecular collections. Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87), Seattle Washington (pp. 590-594).

Crouse, M. and Forbus, K. (2016). Elementary School Science as a Cognitive System Domain: How Much Qualitative Reasoning is Required? In *Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems*. Evanston, IL.

Crouse, M., McFate, C.J., and Forbus, K.D. (2018a). Learning from Unannotated QA Pairs to Analogically Disambiguate and Answer Questions. *Proceedings of AAAI 2018*.

Crouse, M., McFate, CJ., & Forbus, K. (2018b) Learning to Build Qualitative Scenario Models from Natural Language. *Proceedings of QR 2018*, Stockholm.

Dalvi, B., Huang, L., Tandon, N., Yih, W. T., & Clark, P. (2018). Tracking State Changes in Procedural Text: a Challenge Dataset and Models for Process Paragraph Comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*(Vol. 1, pp. 1595-1604).

Das, R., Munkhdalai, T., Yuan, X., Trischler, A., & McCallum, A. (2018). Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. arXiv preprint arXiv:1810.05682.

Drabble, B. (1993) EXCALIBUR: A program for planning and reasoning with processes. *Artificial Intelligence*, 62(1):1-40.

Fan, J., Barker, K., & Porter, B. (2009) Automatic interpretation of loosely encoded input. *Artificial Intelligence*, 173(2):197-220.

Forbus, K. (1984) Qualitative Process Theory. *Artificial Intelligence*, 24, pp. 85-168.

Forbus, K. (1989). Introducing actions into qualitative simulation. *Proceedings of IJCAI-89*.

Falkenhainer, B. and Forbus, K. (1991). Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, 51, 95-143.

Fikes, R. & Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189-208.

Friedman, S., Forbus, K., & Sherin, B. (2017). Representing, Running, and Revising Mental Models: A Computational Model. *Cognitive Science*, 42(4):1110-1145.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155-170.

Hayes, P.J. (1984). The Second Naïve Physics Manifesto. In *Formal Theories of the Commonsense World.* Ablex, Norwood, NJ.

Hogge, J. (1987) Compiling plan operators from domains expressed in qualitative process theory. *Proceedings of AAAI-87.*

Kamp, H., & Reyle, U. (2013). *From discourse to logic: Introduction to model theoretic semantics of natural language, formal logic and discourse representation theory* (Vol. 42). Springer Science & Business Media.

Kiddon, C., Ponnuraj, G. T., Zettlemoyer, L., & Choi, Y. (2015). Mise en place: Unsupervised interpretation of instructional recipes. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 982-992)*.

McFate, C.J., Forbus, K. and Hinrichs, T. (2014). Using Narrative Function to Extract Qualitative Information from Natural Language Texts. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec City, Québec, Canada.

McFate, C., and Forbus, K. (2016). An Analysis of Frame Semantics of Continuous Processes. *Proceedings of the 38th Annual Meeting of the Cognitive Science Society*, Philadelphia, PA, August.

McFate, C., and Forbus, K. (2016). Scaling up Linguistic Processing of Qualitative Process Interpretation. *Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems*. Evanston, IL.

Rickel, J. & Porter, B. (1994) Automated modeling for answering prediction questions: Selecting the time scale and system boundary. *Proceedings of AAAI-94*

Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C., Baker, C., & Scheffczyk, J. (2016) *FrameNet 2: Extended Theory and Practice*. https://framenet2.icsi.berkeley.edu/docs/r1.7/book.pdf

Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. (2017). Bidirectional attention flow for machine comprehension. *Proceedings of the International Conference on Learning Representations (ICLR).*

Tandon, N., Dalvi, B., Grus, J., Yih, W. T., Bosselut, A., & Clark, P. (2018). Reasoning about Actions and State Changes by Injecting Commonsense Knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 57-66).