Advances in Cognitive Systems 4 (2016) Submitted 4/2016; published 6/2016

Scaling up Linguistic Processing of Qualitative Processes

Clifton McFate C-MCFATE@NORTHWESTERN.EDU
Kenneth Forbus FORBUS@NORTHWESTERN.EDU
Qualitative Reasoning Group, EECS Department, Northwestern University, Evanston, IL 60208 USA

Abstract

Many natural language systems either focus on specific domains or sacrifice deep representations for broad coverage. We propose that a combination of a domain independent grammar and semantics along with top-down domain-relevant narrative guidance can achieve both breadth and depth. We investigate one source of top-down guidance in Qualitative Process (QP) theory, a general causal semantics for capturing mental models of continuous processes. Recent work has linked QP models to linguistic frame semantic representations, but to date this work has focused on individual sentences or paragraphs. This paper describes how we have built on and improved representations used in prior work to scale up to chapter-length texts, and to extract complete type level rather than instance-level models. We evaluate our approach using four simplified chapters from a strategy game manual.

1. Introduction

There is an important tradeoff in research on natural language processing between breadth of domain and depth of understanding. Many statistical systems focus on breadth at the expense of depth, operating over large corpora but lacking conceptual representations. Alternatively, many semantic parsers translate from natural language to a domain-specific representation such as a database query language, actions in a GUI, or a robotic control language (Zelle & Mooney, 1996; Branavan *et al*, 2010, Matuszek *et al*, 2013). While useful, these systems don't necessarily extend beyond their initial domains—even to new robotics domains or databases. We argue that a way to achieve both breadth and depth is through the combination of a domain-independent grammar, broad semantic representations[1], and constraints imposed by the current context and task. We express context and task constraints in terms of *narrative functions* (Tomai & Forbus, 2009), which identify functional roles of components of texts. In understanding fables, for example, one function of a sentence might be to introduce a character.

Predicting the function of a sentence in a narrative can be used to constrain the interpretation process and select among competing semantics by making semantic choices that are consistent with the expected function. We model this process as abductive back-chaining. McFate *et al* (2014) argued that constraints from Qualitative Process (QP) theory (Forbus, 1984) are one source of narrative guidance. That is, the function of some sentences (especially in science texts) is to encode or elaborate a QP description of a continuous event.

[1] We use ResearchCyc KB contents, http://www.cyc.com/platform/researchcyc/

QP narrative functions provide a powerful guidance mechanism as continuous phenomena span a vast range of our experience. Examples include boiling water, technical manuals, the dynamics of ecosystems and economics, and even social and mental life (e.g. degree of blame (Tomai & Forbus, 2008)). Naturally, these are discussed in natural language texts. Qualitative, causal models provide a general semantics for capturing mental models of such phenomena, providing coverage for many domains. It has been further argued that QP theory (Forbus, 1984) can provide an inferential semantics for natural language (Kuehne, 2004; McFate *et al, 2014*; McFate & Forbus, 2015).

The incremental nature of language led to reformulating the concepts of QP theory into QP Frames (Kuehne, 2004), which were initially generated by detecting syntactic patterns during parsing. McFate *et al* (2014) created an initial set of narrative functions corresponding to each frame which extracted QP frames from single sentence snippets of advice for a strategy game and models of solar energy from single paragraphs of an elementary science text. While promising, this approach had two problems. First, extracting higher-level frames relied on complete and correct extraction of lower level frames, leading to brittleness. Second, most of the representations produced were instance-level descriptions, i.e. about a specific situation. While concrete examples are common in explanatory texts, much of what is conveyed is more generic, e.g. “water can freeze.” Only type level dependencies were handled in prior work, as opposed to all of QP theory.

This paper describes a new approach, also based on narrative functions, that helps overcome these problems. First, to address brittleness, we reorganized QP frames such that lower-level frames relate through shared lexical reference rather than having one as an argument to the other, thereby supporting fragmentary representations. Second, we address the generic knowledge issue by fully adopting type level QP representations (Hinrichs & Forbus, 2012). Finally, in order to extract these type level representations we introduce new type level narrative functions as well as a secondary model-fragment extraction process. We start by summarizing the key background on QP theory, type level representations, frame semantics, and narrative function driven abduction. Then we discuss our improved QP frames and how they address brittleness in prior work. We give a brief overview of their integration into our current language system. To evaluate them, we describe the results of learning by reading simplified English versions of four chapters of the Freeciv[2] manual. We close with related work, conclusions, and future work.

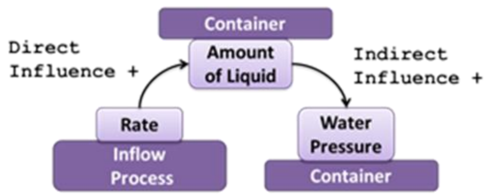
2. Background

2.1 Qualitative Process Theory

In qualitative process theory, changes in a continuous system are the result of processes. Consider water flowing into a container. The amount of water in the container is changing as a consequence of the flow process. Quantities change in QP theory through two different kinds of influences. The rate of a process constrains a quantity through a *direct influence* (represented with the predicates *i+* and *i-* for positive and negative influences). As shown in Figure 1, a direct influence would hold between the rate of flow and the amount of liquid in the container. *Indirect influences* (also called *qualitative proportionalities* or *qprops*) propagate the direct effects of a process through the rest of a system by providing partial information about causal relationships.

[2] Freeciv is an open-source version of Civilization 2, http://www.freeciv.org/

For instance, a qualitative proportionality partially constrains the pressure of water in a container based on the amount of water there is. QP theory also allows for the relative values of quantities to be compared using *ordinal* relationships, and provides *correspondences*, which further constrain quantities when they are qualitatively proportional. A summary of the QP relations we discuss can be found in Figure 1.



Direct Influence: (i+/- <Constrained> <Constrainer>)

Meaning: The constrained increases or decreases as specified by the constrainer

Example: (i+ (LiquidFn tub) (RateFn flow))

"Water flows into the tub."

Indirect Influence: (qprop+/- <Consequent> <Antecedent>)

Meaning: The consequent is qualitatively proportional or inversely qualitatively proportional to the antecedent.

Example: (qprop+ (WaterPressureFn tub) (LiquidFn tub))

"The water pressure in the tub depends on the amount of liquid in the tub."

Ordinal: (qGreater/LessThan/= <Quant1> <Quant2>)

Meaning: Quantity 1 and 2 are >, <, or =

Example: (qGreaterThan (HeightFn tub) (LiquidFn tub))

"The liquid in the tub is lower than the height of the tub."

Correspondence: (correspondence <Quant1> <val1> <Quant2> <val2>)

Meaning: When quantity 1 has value 1, quantity 2 has value 2

Example: (correspondence (LiquidFn tub) (WaterPressureFn tub))

"When the tub is empty, the water pressure is 0."

Figure 1. Summary of QP Primitives and Example Model for Water Flow

To explicitly represent the conditions under which domain knowledge is applicable and relevant, QP theory uses *model fragments*. A model fragment is a logically quantified description that expresses an aspect of an object, process, or concept. They can be thought of as a kind of schema. Each model fragment specifies *participants*, logical variables which must be bound to instantiate that fragment for any particular scenario. *Constraints* among the participants provide further guidance on when instantiating that fragment would make sense. Model fragments also have conditions and consequences. When the *conditions* of a model fragment hold, then the *consequences* that fragment imposes among its participants hold. For example, Figure 2 describes a model fragment for contained liquid which specializes a more general notion of contained stuff with the properties it has by virtue of being a liquid. It has three participants, ?stuff, which denotes the contained stuff, which is constrained to be in the liquid phase, ?sub, the material it is made out of, and ?can, the container which defines it. When an instance of this model fragment is active, it means that there is an indirect influence (qprop) between the pressure of the contained liquid and its amount – since it is positive, an increase in the amount will cause an increase in the pressure, all else being equal.

The compositionality of model fragments means that the causal structure for a specific situation is assembled by gathering all of the relevant model fragments (hence the "all else being equal" qualifier above, since a closed world assumption is required). This means that a domain theory made of general model fragments can be used to create models for a wide range of situations.

```
(defModelFragment ContainedLiquid
   :participants ((?stuff :type ContainedStuff
                         :constraints (phaseOf ?stuff Liquid))
                        (?sub :type Substance
                           :constraints (substanceOf ?stuff ?sub))
                        (?can :type Container
                           :constraints (containerOf ?stuff ?can)))
   :conditions   ((active ?stuff))
   :consequences ((qprop (Pressure ?stuff) (AmountFn ?stuff))
```

Figure 2. Model Fragment for Contained Liquids

2.1.1 Type Level Representations

Traditional qualitative reasoning instantiates model fragments to assemble instance level, or propositional, models of situations. Our examples in Figure 1 describe an individual instance of a container. Type level representations were created because for many tasks instance-level models are impractical. An example domain where type level representations have proven necessary is the strategy game Freeciv, where players build a civilization by exploring, creating cities, improving cities and terrain, and researching new technology. Strategy games are a good test bed for type level reasoning because they require balancing economic concerns, short-range goals versus future investments, and conducting military operations. Planning the expansion of a civilization with new cities requires reasoning about units that do not yet exist. Moreover, such games, like real life, contain more individuals than it is feasible to reason about in full detail all at once. Hence more focused reasoning is required, which type level representations support. In terms of learning by reading, type level representations have another advantage: Many texts, especially explanatory texts, describe processes and causality in general terms rather than with instantiated entities. This includes science books and strategy manuals.

Hinrichs and Forbus (2012) present a QP formalism for encoding type level models. These type level QP representations differ from standard QP relations in that their arguments are collections[3] and predicates rather than individuals. Type level influences are of the form:

`(i, qt1, qt2, c1, c2, r)` where:

- `i =` a second-order influence predicate
- `qt1, qt2 =` quantity types (represented as denotational functions)
- `c1, c2 =` collections for the entity arguments to the quantity types
- `r =` a binary relationship that holds between the instances of c1 and c2

As an example, the following type level direct influence states that a direct influence holds between the tax rate of a city and the gold of the player given that a player owns a city.

```
(i+TypeType (MeasurableQuantityFn currentGold)
  (MeasurableQuantityFn currentTax) Freeciv-Player Freeciv-City owner)
```

[3] That is, unary predicates denoting concept membership, in Cyc terminology. They can be thought of as classes or types of things.

Hinrichs and Forbus also introduce positive and negative dependency predicates that describe the influence of non-quantity relationships, such as the effect of building a granary in a city in Freeciv, or improving a land tile to produce more resources for a city. The following statement captures the semantics of "Irrigating a tile improves food production."

```
(positivelyDependsOn-TypeType
  (MeasurableQuantityFn tileFoodProduction)
   FreecivLocation FC-Special-Irrigation specialAt)
```

Model fragments can also be defined using type level predicates, to avoid explicit quantification. The underlying semantics is the same. For example, `participantType` associates a fragment with a required role relation and the kind of entity that can fulfill that role. The destination of a liquid flow process might be represented like so:

```
(participantType LiquidFlowProcess toLocation LiquidContainer)
```

Constraints and conditions are represented with the `participantConstraint` and `conditionOf-TypeType` predicates which add additional requirements to the entities specified by the role relation in the participant assertions. Consequences are represented using the `consequenceOf-TypeType` predicate which specifies that an influence holds for all entities which satisfy the participant conditions. For liquid flow, a consequence might be that the liquid in the destination is increased based on the rate of flow (a direct influence), i.e.

```
(consequenceOf-TypeType LiquidFlowProcess
 (i+  ((QPQuantityFn AmountFn) (LiquidContentsFn toLocation))
      ((QPQuantityFn Rate) processInstanceOf))
```

McFate *et al* (2014) showed that a half-dozen sentences of natural language advice, whose semantics were captured by type level influences, sufficed to enable a Companion (Forbus et al, 2009) to improve performance in Freeciv. But what about learning larger scale models by reading, i.e. model fragments? The incremental and ambiguous nature of language makes this difficult. For example, consider how the information from the same model fragment might be split across sentences in English:

- Heat flows from a hot object to a cool object.
- Heat will flow from a hot object. The heat flows to a cold object.
- Heat will flow between objects. This occurs as long as the source has a lower temperature than the destination.

Sometimes the information that one expects in a subsequent sentence never comes. For example, "Citizens consume food." From where? Do they get food from the land, from their city, from the entire civilization? Domain knowledge is needed to infer or hypothesize the answer. This suggests using a representation that supports incrementality as an intermediate stage in learning qualitative representations from text, leaving extraction of full model fragments to a later stage. Frame semantics provides exactly this kind of representation, so we examine it next.

2.2 Frame Semantics and QP Frames

Frame semantic approaches link lexical representations to conceptual schemas called semantic frames. Fillmore *et al*'s (2001) FrameNet is a frame semantic resource for English. In FrameNet, a frame is evoked by a lexical unit (word) in a specific syntactic construction called a *valence pattern*. For a construction, the frame uses the arguments of the sentence to fill frame-specific semantic roles called *frame elements*. Frame elements can be thought of as binary role relations that relate a word or phrase to its role in a conceptual schema.

For example, the `Motion` frame defines frame elements for the `Source`, `Goal`, and `Theme` (the thing moving). It is evoked by a lexical unit such as the word *go* in "The boy went to the store." Here, the noun phrase (NP) subject fills the role of `Theme` and the prepositional phrase (PP), 'to the store', fills the role of `Goal`. This specific ordering of phrases is a valence pattern. Thus, for a frame like `Motion` FrameNet provides the frame elements of a motion event, lexical units that evoke motion, and annotations of how specific valence patterns align with these frame elements.

Kuehne (2004) provided an initial mapping between QP theory relations (as in figure 1) and FrameNet style linguistic frames. Kuehne's formulation begins with quantity frames which act as arguments to higher-order influence frames. Just as traditional FrameNet frames are evoked by a lexical unit in an instantiating valence pattern, quantity frames are evoked by a quantity evoking unit (such as the word *heat* or *temperature*) in a supporting syntactic pattern (e.g. a possessive: *The brick's mass*). A quantity frame has the core elements `quantityType`, `quantityVar`, and `entity`. `QuantityVar` is the quantity evoking lexical unit. The `entity` is the lexical unit that the quantity pertains to and the `quantityType` describes the collection the quantity belongs to.

Quantity frames act as arguments to influence frames. Direct influence frames take a `constrained` and `constrainer` quantity frame, and similarly, indirect influences take a `consequent` and `antecedent` quantity. Both frames have a `sign` element for the direction of change. A similar formalism is used in McFate *et al* (2014). However, this prior work was not intended to capture type level QP models. Furthermore, this approach suffers from the shortcoming that constructing influence frames depends on the complete specification of quantity frames, leading to brittleness. As an example, consider the previously discussed statement from the Freeciv domain:

"Citizens consume food."

As we previously noted, this sentence leaves out the required `entity` role, what FrameNet calls a null instantiation. This kind of construction (a causative) reveals an important constraint on the process, that it is performed by citizens. However, because it does not fully specify a quantity frame, the previous approach did not construct a direct influence. While it is possible to specify direct influence rules for incomplete quantity frames, different process types affect what kinds of information can be linguistically null instantiated in different ways. Another case to consider is when the quantity itself is not recognizable as such. For example:

"Citizens produce phlebotinum."

A system not well versed in fictional substances would have difficulty recognizing phlebotinum as a substance, which implies a quantity type, but its morphology suggests a substance and the verb and syntax suggests that some quantity of it is created in a production process. One benefit of mapping to FrameNet is that its valence patterns tell us the roles of arguments to process verbs, enabling the inference of an influence even with incomplete information.

In extending QP frames for type level representation we address brittleness by changing the formalism such that QP frames are related through shared lexical units rather than having frames act as arguments to one another. This allows more flexibility in describing partially completed models. We discuss these changes in more detail in section 3. Now, we move to narrative function and its role in extracting these QP frames from text.

2.3 Narrative Function Abduction

Kuehene's (2004) approach to recognizing quantity frames relied on specific syntactic patterns. McFate *et al* (2014) expanded on this work by incorporating QP frame detection into a narrative function driven language interpretation system. Narrative functions are a way of generating expectations in reading as well as checking comprehension. They tie individual sentences to their role in a broader discourse (Labov & Waletzky, 1966; Barthes 1977; Trabasso et al. 1984; Tomai & Forbus, 2009). Introducing a character is a narrative function, as is introducing an event and raising expectations about possible outcomes. Tomai & Forbus (2009) showed that narrative functions could be used in understanding fables. The goal of narrative functions is to contextualize semantic interpretation for the results of a broad, domain-independent grammar.

Narrative functions operate over potential semantic interpretations to disambiguate in favor of contextually relevant meanings. For example, consider the phrase “the hot brick”. The word *hot* is ambiguous. It could be that the brick has a high temperature, is very physically attractive, or is even experiencing the sensation of being hot. The context of processing (a logical environment specified via Cyc microtheories) provides access to narrative function rules that detect meanings relevant to that context. When reading a science text, for instance, introducing a temperature quantity is more likely to be relevant and so the lexical and parse choices that lead to that interpretation are selected. Conceptually, we view narrative functions as detectors that select for specific kinds of information.

In McFate *et al* (2014) narrative function abduction is implemented via back-chaining, where narrative functions are the goal, with lexical and syntactic choices allowed to be assumed. Given a set of possible semantic interpretations and a domain, the system queries for relevant narrative functions in that domain. It then selects interpretations that are consistent with the expected narrative functions, maximizing how much of the text can be explained (see Section 4).

In the following sections we present type level QP frames, a formalism intended to capture type level semantics and to address the brittleness of previous approaches. We then describe how we extract these new frames from text using the narrative function abduction process. We conclude with an analysis of coverage over several simplified chapters from the Freeciv manual.

3. Type Level QP Frames

First we describe how we have modified and expanded QP frames to represent type level representations like those in section 2.1.1. While Kuehne’s (2004) influence frames took quantity frames as arguments, we instead choose to have frames relate through their shared lexical units. This is closer to FrameNet’s representation and facilitates finding influence frames even when the system fails to infer quantity frames or only infers only partial information about them. We discuss, in order, *quantity frames*, *direct influence frames*, *indirect influence frames*, *ordinal frames*, *type level dependencies*, *participant state frames*, and *model fragment frames*.

Like their instance level counterparts, type level quantity frames have an `entity`, `quantityType`, and `quantityVar` frame element. However, they also have an additional `entityType` element for the collection of the entity. In the type level model fragments, these collections act as arguments rather than the individuals. In the sentence, “A citizen consumes food points from the city”, food-points would have the following type level quantity frame:

```
QP Frame TypeLevelQuantityFrame13964:
    entity: city
    entityType: FreeCiv-City
    quantityType: (resultingQTypeFn FoodPoints)
    relatesToQTypeVar: food-points
```

To adapt our influence representation to take lexical units instead of frames, we break apart the constrained and constrainer arguments into constrained and constrainer quantity types and entities. For example, "A citizen consumes food points from the city", would lead to the direct influence:

```
QP Frame TypeLevelInfluenceFrame8926:
     agentiveCauser: citizen
     constrainedQType: (resultingQTypeFn FoodPoints)
     constrainedQuantity: food-points
     constrainedEntity: city
     constrainedEntityType: FreeCiv-City
     constrainerEntity: consume
     constrainerQType: Rate
     sign: -1
```

The type level influence frame is related to the quantity frame through their shared quantity lexical unit, *food-points*. Unlike our previous approach, even if the quantity frame lacked an entity (as in just "A citizen consumes food.") the influence frame would still govern the quantity frame through the shared quantity lexical unit. Here, the entity can be null instantiated. We also introduce a new frame element, `agentiveCauser` which is used to capture events which have a causal participant. This allows us to capture requirements such as that a citizen be the one consuming even though the citizen itself is not involved in its own quantity frame. Previously, no agent role existed.

Type level indirect influence frames take an `antecedent` and a `consequent` quantity as well as a `sign`. Again, shared linguistic variables connect indirect influences to their quantity frames instead of the frames themselves being arguments. Below is the indirect influence produced for the sentence, "the food required in a city depends on the size of the city", as well as the quantity frame for required food.

```
QP Frame TypeLevelQPropFrame3197:
    antecedentQuantity: food
    consequentQuantity: size
    sign: 1
QP Frame TypeLevelQuantityFrame15647:
    entity: city
    entityType: FreeCiv-City
    quantityType: (AmountRequiredFn food)
    relatesToQTypeVar: food
```

Type level ordinal frames take three arguments, two quantities and an ordinal relation which holds between the quantities. Like influence frames, ordinals relate to quantity frames through shared lexical arguments.

We also introduce *participant state frames* which specify that an individual in a process has a constraining state of a particular type. As an example, settlers in Freeciv consume different amounts of food under different governments; hence government type is a constraining state. Previously, QP frames did not capture linguistic constructions for prerequisite entity states, only conditional constructions and ordinals.

This new formalism addresses the shortcomings of prior work in two key ways. First, it has been extended to include necessary type (collection) information as frame elements (e.g. entityType). This allows for type level model fragments as discussed below. Second, relating frames through shared lexical units instead of making quantity frames a sub-frame of influence frames reduces brittleness by allowing partial descriptions of influences in the face of lexical ambiguity.

3.1 Model Fragment Frames

While McFate *et al* (2014) extracted individual type level dependency statements, going directly to statements sacrificed the incrementality of frames, and did not handle the rest of QP theory. Here we introduce model fragment frames as a means of collecting QP frames extracted from individual paragraphs and applying them to form a complete model. Unlike influence frames, model fragment frames do take other frames as their arguments. Each sub-frame is a relation in the model fragment. For example, `participantType` has a corresponding participant frame. Each participant frame has an entity which is the lexical word for the participant and a constraining role. The constraining role is the binary predicate that relates the entity to the process event (e.g. source / destination). Additional participant constraints are represented as a frame element for the participant frame. Model fragment consequences are represented with a consequence frame which has two quantity arguments and an influence relation. Consequences and participants are related through participant and consequence roles to a model fragment frame. Model fragment frames can have a condition frame element to represent activation conditions. Figure 3 shows a set of model fragment frames for the sentence: “Heat flows from the brick to the ground because the temperature of the brick is greater than the temperature of the ground”.

```
QP Frame FluidFlow-Translation601691:
 condition: (greaterThan
             ((QPQuantityFn Temperature) from-UnderspecifiedLocation)
             ((QPQuantityFn Temperature) to-UnderspecifiedLocation))
 consequence: ConsequenceFrame601695
 participantFrame: ParticipantFrame601693
 participantFrame: ParticipantFrame601692
 processEvoker: flow
 processType: FluidFlow-Translation
QP Frame ConsequenceFrame601695:
 consequenceFrameArg1: ((QPQuantityFn ThermalEnergy) from-UnderspecifiedLocation)
 consequenceFrameArg2: ((QPQuantityFn Rate) processInstanceOf)
 relation: i-
QP Frame ParticipantFrame601692:
 entity: ground
 role: to-UnderspecifiedLocation
QP Frame ParticipantFrame601693:
 entity: brick
 role: from-UnderspecifiedLocation
```

Figure 3. Example of Model Fragment Frames

4. System Overview

Figure 4 provides a high-level overview of the flow of processing in our language system. EANLU (Tomai & Forbus, 2009), uses Allen’s (1994) chart parser and Grishman *et al*’s (1993) COMLEX lexicon. A rule driven feature-based grammar builds syntactic forms and unifies them with neo-Davidsonian[4] semantic templates from ResearchCyc. These templates are explicitly

[4] In neo-Davidsonian semantics an event is reified as an individual that binary role relations (frame elements) can relate to. For example, in “the boy eats”, eat would be represented as an eating event with the agent being the boy.

linked to individual lexical units in COMLEX. The parser represents syntactic and semantic ambiguity using disjunctive choice-sets. Returning to “the hot brick”, each semantic meaning for *hot* is represented within a disjunction from which an individual meaning can be selected and propagated. In Figure 4 we show example semantic choices for the brick’s temperature and its physical sensation (as in it feels hot). Once the set of choice-sets is produced, the system analyzes the current sentence and discourse level contexts for narrative functions.

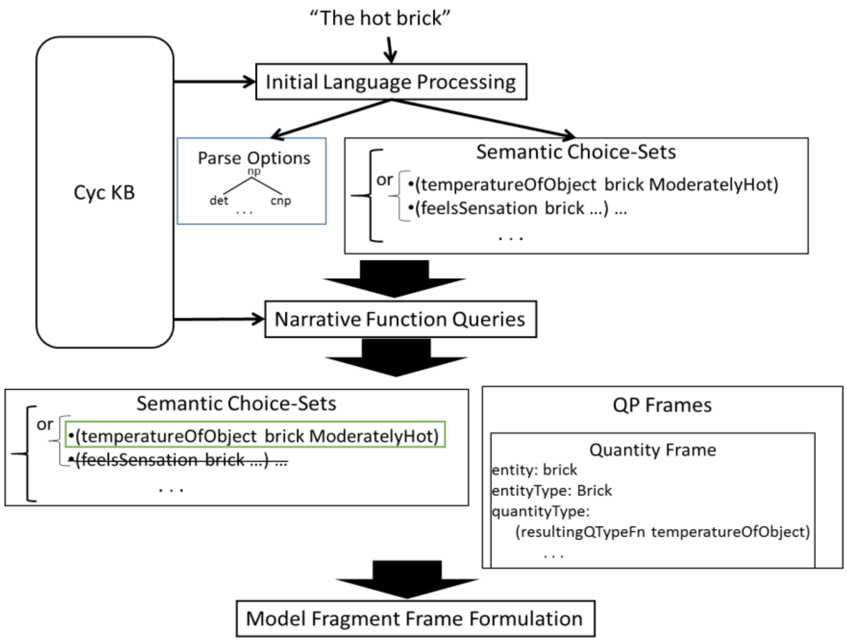


Figure 4. Information flow in the system

Narrative functions available in the current logical environment are queried for incrementally and are ordered such that lower-level functions like quantities can be found before higher-level ones like influences.

Each narrative function can be proven true given a set of Horn clause rules that trigger given a QP relevant semantic interpretation. Choices that are consistent with these interpretations can be abductively assumed, allowing for goal-driven disambiguation. For example, in Figure 4 only `temperatureOfObject` is consistent with a quantity frame interpretation. Conflicts can be resolved by using weighted heuristics that prefer certain expressions over others. We can exert more direct control by nesting abductive queries within the narrative function horn clauses, forcing certain sub-queries of the system to make choices before others are allowed to continue. This allows us to effectively define decision points so that future queries can rely on already assumed facts. This is also useful for speeding up the system by constraining the search space of following narrative functions. One drawback is that our current abductive mechanism does not allow backtracking. Thus, forcing a decision point can result in making choices that are locally optimal for one narrative function but potentially inhibit a more complete understanding.

The narrative function mechanism is also able to make use of a set of reference resolution heuristics called on lexical units in relevant expressions. Example heuristics include resolving pronouns based on gender, preferring reference to sentence subjects, and allowing reference between nouns with the same orthographic form or semantic collection.

For each paragraph, after the initial set of QP narrative function queries have been completed, a secondary rumination process (Forbus et al, 2007) builds model fragment frames based on the current paragraph interpretation. This differs from previous approaches in that, in both Kuehne's (2004) and McFate *et al*'s (2014) systems, process frames were extracted at the same time as quantities and influences. This change is motivated by the observation that model formulation can and often does rely on domain-specific modeling assumptions and significant world knowledge that is beyond the scope of the linguistic model. By separating out this process we allow future work on model learning and integration to proceed in parallel with linguistic interpretation.

5. Experiment

To evaluate our system we extracted type level frames and model fragments using the new type level frame implementation for learning by reading. Our corpus was four simplified chapters from the Freeciv manual:

- Economics (115 sentences): describes managing cities, food production and consumption, producing and using gold, the effect of luxuries on citizen happiness, and trade.
- Cities (60 sentences): describes building cities, working terrain, adding buildings, and citizen management, and city disorder relative to citizen happiness.
- Units (partial) (34 sentences): describes different kinds of units, movement points, unit actions, and zones of control. (The equivalent of tabular data was omitted, since it would inflate the statistics.)
- Combat (50 sentences): describes attacking, hit points, healing, and military unit types.

The simplification process is that of Barbella and Forbus (2011). Syntax is simplified by breaking complex sentences into multiple sentences, but leaving the vocabulary intact when possible. We also reduced anaphoric reference and made some pragmatically implied arguments explicit. As an example: "A city produces food. Citizens consume food" would be turned to "A city produces food. Citizens consume food from the city."

Overall, the system found 139 QP frames and constructed 69 frames for model fragments from the interpretations. No model fragment frames were found in the combat chapter. These results are summarized in Tables 1 and 2. The majority of frames were found in the economics chapter which focused on city and civilization-level processes (rather than those of individual units).

Table 1: QP Frames by Chapter

Chapter	Quantity Frames	Qprop Frames	DI Frames	Participant State Frame	Ordinals
Economics	58	15	17	5	0
Cities	18	0	5	0	0
Units	8	2	3	0	0
Combat	8	0	0	0	0

Table 2: Frames for Model Fragments by Chapter

Chapter	Model Fragment Frames	Model Fragment Participants	Model Fragment Consequences	DependsOn-TypeType
Economics	25	17	17	2
Units	3	0	2	0
Cities	2	1	0	0

The system was most successful at creating models of food consumption. Figure 5 shows a model fragment for food consumption by a settler under anarchy. There are other model fragments for settler food consumption corresponding to different government types, which were correct except that it conflated two different government conditions, i.e. democracy and republic.

While correct, a more efficient representation would specify a single food consumption process and use model fragments conditioned on different government types to provide additional constraints on the rate. This kind of non-local optimization seems best performed during later processing, such as rumination.

```
QP Frame DestructionEvent1200721:
 consequence: ConsequenceFrame1207285
 consequence: ConsequenceFrame1207284
 participantFrame: ParticipantFrame1200730
 participantFrame: ParticipantFrame1200729
 processEvoker: consume
 processType: DestructionEvent
QP Frame ConsequenceFrame1207284:
 consequenceFrameArg1: ((QPQuantityFn (resultingQTypeFn FoodPoints))
                                  from-UnderspecifiedLocation)
 consequenceFrameArg2: ((QPQuantityFn Rate) processInstanceOf)
 relation: i-
QP Frame ConsequenceFrame1207285:
 isa: ConsequenceFrame
 consequenceFrameArg1: ((QPQuantityFn Rate) processInstanceOf)
 consequenceFrameArg2: ((PerFn FoodPoints Turn-GameEvent) 1)
 relation: q=
QP Frame ParticipantFrame1200729:
 entity: settler
 participantconstraint:(underInfluenceOf settler1198192 anarchy1198183)
 role: doneBy
QP Frame ParticipantFrame1200730:
 isa: ParticipantFrame
 entity: city
 role: from-UnderspecifiedLocation
```

Figure 5. Frame semantic structure for
"Under anarchy, a settler consumes one food point per turn."

The system found individual influences for several other processes. For example, it learned two direct influences representing a city's accumulation of food and production points. It also identified that cities can convert trade-points to gold, finding a positive direct influence for gold and a negative direct influence for trade-points attached to the same conversion event. In this case, we fail to get a full model fragment because the system isn't able to tell whether the gold or trade is global or specific to the city.

The system also extracted indirect influence (qprop) relationships, though these were frequently partial, lacking a connecting quantity frame for one or both quantities. Of the 17 total qprops found, only 4 had both antecedent and consequent corresponding quantity frames. Several errors came from mistakes in reference resolution. Several other errors were due to dependency relationships with a state rather than a specific quantity (e.g. "Corruption depends on the type of government") which is not specific enough for a positive or negative type level dependency.

We evaluate recall at the level of QP influence frames (direct and indirect influence frames). Judging overall recall is difficult, as differences in linguistic representation and domain assumptions could result in models that capture different versions of the same processes. Since frames are interconnected, reference failures can result in an incomplete model even if the individual frames are correct. Since capturing partial information was part of the goal of QP frames, we want to evaluate partial models. We first compare to a hand-made count of expected QP frames (not including model fragment level frames) across all four chapters in order to evaluate linguistic pattern coverage. We identified 279 possible frames, 195 quantity frames, 51 DIs, 19 qprops, 5 ordinals, and 9 participant state frames. The system identified 139 frames. For this evaluation, we counted a frame as correct if it had all correct required frame elements. Influences missing a corresponding quantity frame (through reference failure or error) were counted as incorrect. A quantity frame was incorrect if there was no quantity frame of the same type in the manual annotation or if any individual assertion (e.g. entityType) was not supported by the text. By these measures we found 40 incorrect or incomplete frames out of the 139. This gives us an overall recall of .35 and a precision of .71. While low, this doesn't include partial models and many missing frames were unconnected quantity references in the text. A better measure is to evaluate sets of complete or partial representations of influences and their connecting frames. We evaluated our influences and their connecting frames at the level of correct assertions. If an influence had a correct variable for a quantity, but that quantity lacked a frame, the individual assertion was correct but we wouldn't have the assertions from the missing frame. Across the entire corpus, we expected 950 assertions for the 66 influences and their associated frames. On this subset, the system produced 412 correct assertions and 56 erroneous assertions for complete or partial models. This results in a recall of .42 and a precision of .88

An analysis of the documents also revealed several places where QP frames would not currently be generated but where qualitative information could be extracted. Especially in the Freeciv manual, continuous processes are discretized into turns. These individual statements can be used to form a qualitative representation, but QP frames do not currently capture these step-wise descriptions. We suggest how they may be adapted to do so in future work.

6. Discussion

Using narrative functions to extract type level frames, the system found several type level models and many more partial models in the Freeciv manual. Specifically, models of consumption, production, and conversion provide potentially valuable information about trade-offs in city management. However, clearly multiple challenges remain. One is improving reference resolution. In our new approach, frames are related through shared lexical units, thus complex descriptions rely on accurate co-reference of lexical units across sentences. Our system currently does not recognize metonymic reference (e.g. referring to a car as "a set of wheels"). Another class of reference problems is reference of synonymous complex expressions (e.g. "population" vs "number of citizens" vs "the people living in…"). A third difficulty arises in connecting verbal descriptions of processes to their nominalized form (e.g. "Heat flows" followed by "the heat flow facilitates…"). Another source of errors came from our use of decision points within narrative function sub-queries. As discussed in section 4, decision points allow us to enforce assumptions from lower-level narrative functions and limit the search space of subsequent narrative functions. However, these choices are not necessarily globally optimal. Instead, in future work we intend to accumulate proof paths for narrative functions of the same level and then use dependency directed search to find the set of choices that maximizes the number of narrative functions identified. This should significantly improve our ability to build consistent frames.

7. Related Work

The most closely comparable work would be that of Branavan *et al.* (2011) which used dependency parses of the Civilization 2 manual to link together game concepts and influence a Monte Carlo learner, which then played a version of the game. However, their version was so limited -- 1/4th the size of the default game board and ending games at 100 turns – that most of the game's complexities are factored out. Moreover, their system used the game engine itself to do massive look-ahead computations. Instead, in prior work (McFate et al. 2014) we have used qualitative models learned from relatively little linguistic input to influence game strategies.

Our approach to narrative function abduction draws on Hobbs (2004) but differs in that our abductive assumptions are constrained only to choice sets. While Hobbs relied on a mathematical cost function to weight abductive assumptions, our system relies on type level knowledge from the Cyc ontology. Ovchinnikova (2012) also used abductive reasoning to produce frame semantic representations from text, however their approach relied on lexical knowledge to weight abductive inferences rather than type level top-down guidance. However, certainly lexical level weighting could be applied to our system in the future.

While our system operates over Cyc representations and lexical information from Grishman *et al*'s (1993) Comlex, recently there has been broader interest in large semantic lexicons. An example is Allen's (2014) semantic-lexicon. To construct it, they extended an existing semantic-lexicon by first using WordNet's synset hierarchy to map unknown words to known ontological concepts and create initial lexical entries. They then extend these entries using parses of the word gloss and WordNet examples to better specify an ontological parent-class and to extract argument structure and lexical entailments. Such research is complementary to our work, and certainly our performance could improve with broader semantic coverage. In turn, narrative function provides a way of constraining possible interpretations from this broad resource.

Finally, our work could also benefit from related work in frame semantic parsing such as Das *et al*'s (2014) SEMAFOR program. Such systems use statistical techniques to annotate text with FrameNet frames. As we continue incorporating FrameNet representations into our system, such techniques could provide a source of evidence for disambiguation.

8. Conclusions & Future Work

We have argued that narrative functions provide a powerful mechanism for top-down guidance, helping to achieve depth within a domain without sacrificing breadth of coverage. We've also argued that Qualitative Process theory acts as a powerful source of narrative functions. While previous work mapped QP theory to linguistic representations (frame semantic frames), it did not extend to type level representations. Furthermore, prior approaches relied on accurate recognition of quantity frames in order to specify influences. This made the system brittle in the face of incomplete information, something frame semantic representations should specifically help with. In this paper we address these shortcomings, first by implementing a new type level QP frame and model fragment frame representation, and then by incorporating these new representations into a narrative-function driven interpretation process. We evaluate our new approach on a corpus of four simplified English chapters from the Freeciv manual, and find that, while work remains to be done, our approach is useful in extracting qualitative models from text. Our approach was particularly successful at recognizing descriptions of production, consumption and conversion.

We see three lines of future work as important. In complex domains, people's thinking and their language often heavily intermingle discrete and continuous perspectives. This holds for Freeciv as well, as illustrated by this description of city growth:

> *"When the amount of food stored in a city becomes full, the population grows by one citizen. The growth causes the amount of food stored in the city to become empty. Building a granary in a city increases its growth rate. With a granary in a city, the amount of food stored becomes half-full after growth. The amount of food stored in a city needed to reach full depends on the population of the city. This means that each new citizen is more costly than the last."*

At the level of turns, there are specific increments added to the food storage. Abstracting to a continuous model seems natural, since we often think in spans of time longer than turns. But when a limit point is reached, a discrete change occurs. Sometimes the effects of a limit being reached are described more procedurally, e.g. the order in which types of units die when a city is gripped by starvation. Descriptions of processes in game explanations often switch back and forth between discrete and continuous perspectives (e.g. "Repair restores 1 hit point per turn." "Building something costs production points."). Similar shifts in thinking happen in other fields, e.g. abstracting discrete sales in a business into a continuous rate at higher levels of abstraction. Thus this is a problem that is definitely worth investigating, and adapting continuous representations to discrete descriptions like these will be an area of future work. We suggest a schema-based approach, but capturing the lexical features of these descriptions could require altering QP frames.

The second line of work involves incorporating domain knowledge into the model frame formulation process. The third is to test these narrative functions in a broader range of domains. Ultimately, we argue that QP theory provides a powerful source of causal reasoning, but it will have to integrate with other broad-coverage modules to fully interpret text.

References

Allen, J. F. (1994). *Natural language understanding (2nd Ed.)*. Redwood City, CA: Benjamin/Cummings.

Allen, J. F. (2014). Learning a Lexicon for Broad-Coverage Semantic Parsing. In the Proceedings of the ACL 2014 Workshop on Semantic Parsing. Baltimore, MD.

Barbella, D. and Forbus, K. (2011). Analogical Dialogue Acts: Supporting Learning by Reading Analogies in Instructional Texts. *Proceedings of AAAI 2011*, San Francisco, CA.

Barthes, R. (1977). *Introduction to the structural analysis of narratives (S. Heath, Trans.). In Image, music, text*. New York: Hill and Wang.

Branavan, S., Silver, D., & Barzilay, R. (2011). Non-Linear Monte-Carlo Search in Civilization II. *Proceedings of IJCAI 2011*.

Branavan, S. R. K., Zettlemoyer, L. S., & Barzilay, R. (2010). Reading between the lines: Learning to map high-level instructions to commands. *Proc. of ACL-2010*. Uppsala, Sweden.

Das, D., Chen, D., Martins, A. F. T., Schneider, N., & Smith, N.A. 2014. Frame-Semantic Parsing. *Computational Linguistics*, 40(1), 9-56.

Fillmore, C. J., Wooters, C., & Baker, C. F. 2001. Building a Large Lexical Databank Which Provides Deep Semantics. *Proceedings of PACLIC-2001,* Hong Kong, China.

Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence*, *24*, 85–168.

Forbus, K, Klenk, M. and Hinrichs, T. (2009). Companion Cognitive Systems: Design Goals and Lessons Learned So Far. *IEEE Intelligent Systems*, 24(4), 36-46.

Forbus, K. D., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., & Ureel, L. (2007). Integrating natural language, knowledge representation and reasoning, and analogical processing to learn by reading. *In Proc. of AAAI-2007*. Menlo Park, CA.

Grishman, R., Macleod, C., & Wolff, S. (1993). *The COMLEX Syntax Project*. Ft. Belvoir: Defense Technical Information Center

Hinrichs, T. and Forbus, K. (2012). Toward Higher-Order Qualitative Representations. *Proceedings of the 26th International Workshop on Qualitative Reasoning*. LA, CA.

Hobbs, J.R. (2004). *Abduction in Natural Language Understanding*. In Horn, L. and Ward, G. (eds.), Handbook of Pragmatics, pp. 724-741, Blackwell.

Kuehne, S. E. (2004). On the Representation of Physical Quantities in Natural Language Text. *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, Chicago, IL.

Labov, W., & Waletzky, J. (1966). *Narrative analysis: Oral versions of personal experience*. In J. Helms, (Ed.), Essays on the Verbal and Visual Arts, pp. 12-44, Univ. of Washington Press.

Matuszek, C., Herbst, E., Zettlemoyer, L., & Fox, D. (2013). Learning to parse natural language commands to a robot control system. *In Experimental Robotics* (pp. 403-415). Springer International Publishing.

McFate, C.J. and Forbus, K. (2015). Frame Semantics of Continuous Processes. *Proceedings of the 28th In-ternational Workshop on Qualitative Reasoning*. Minneapolis, MN, USA.

McFate, C.J., Forbus, K. and Hinrichs, T. (2014). Using Narrative Function to Extract Qualitative Information from Natural Language Texts. *Proceedings of AAAI 2015*, Québec City, Canada.

Ovchinnikova, E. (2012). *Integration of World Knowledge for Natural Language Understanding*. Atlantis Press, Springer.

Tomai, E., Forbus, K. (2008). Using Qualitative Reasoning for the Attribution of Moral Responsibility. *Proceedings of COGSCI 2008.* Washington, D.C.

Tomai, E. and Forbus, K. (2009). EA NLU: Practical Language Understanding for Cognitive Modeling. *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*. Sanibel Island, Florida.

Trabasso, T., Secco, T., & van den Broek, P.W. (1984). *Causal cohesion and story coherence*. In H. Mandl, N.S. Stein & T. Trabasso (Eds.), Learning and comprehension of text (pp. 83- 111). Hillsdale, NJ: Lawrence Erlbaum Associates.

Zelle, J., & Mooney, R. (1996). Learning to parse database queries using inductive logic programming. In *Proc. of AAAI*, pp. 1050-1055. Portland, OR.