Two Kinds of Paraphrase in Modeling Embodied Cognitive Agents

Marjorie McShane, Sergei Nirenburg, Stephen Beale

Institute of Language and Information Technologies
University of Maryland Baltimore County
{marge, sergei, sbeale}@umbc.edu

Abstract

Embodied cognitive agents must be able to process two very different kinds of inputs: interoceptive stimuli (stimuli originating inside of the body) and natural language. To support these complex functionalities, developers must address, among other issues, two kinds of paraphrase: a) reformulation of the representation of physiological events (e.g., symptoms) in "lay" terms that can be understood and remembered by the cognitive agent and b) representation of the meaning of verbal messages in terms compatible with how related content is stored in the cognitive agent's memory. In this paper we describe how these types of paraphrase are treated within a system that uses the same knowledge representation language throughout while permitting different agents to have idiosyncratic ontologies and fact repositories (memories).

Introduction

We understand embodied cognitive agents to be "double agents" consisting of a physiological agent (PhysA), which models the body and associated involuntary physiological events, and the cognitive agent (CogA), which models the perception, reasoning abilities and voluntary actions of the agent. We have implemented such an agent in *Maryland Virtual Patient* (MVP)[1], an agent-oriented simulation and tutoring system. In MVP, a human user plays the role of a physician in training who must diagnose and treat a virtual patient – implemented as an embodied cognitive agent – over time, without recourse to canned scenarios, and with or without the help of a virtual mentor agent (see, e.g., McShane et al. 2007). The physiological agent "lives" over time and responds in realistic ways to disease progression and medical interventions. The cognitive agent experiences symptoms, decides when to consult a physician, makes decisions about its lifestyle, treatment options, etc., and communicates with the human user using natural language.

As should be clear even from this brief overview, MVP is a knowledge-intensive application. The processing in both simulation and NLP is currently supported by hand-crafted knowledge. The core element of this knowledge is a general-purpose ontology that includes particularly well-developed medical knowledge. All ontological OBJECTs and EVENTs are described using a large number of RELATIONs and ATTRIBUTEs. EVENTs in the ontology can be quite complex, containing subevents at many levels and representing causal and temporal chains and scripts (using the terminology of Schank and Abelson 1977). Complex events describe disease progression and treatment, the plans and goals of patients and physicians, clinical practices, rules of dialog in general and medical interviews specifically, and so on. In MVP, the "complete" ontology is used only by the mentor agent. A subset of this ontology, from which the knowledge of best clinical practices has been removed, is used as the ontology of PhysA. A subset of this latter ontology, with some or all medical knowledge removed, is used as the ontology of CogA.

The virtual mentor and CogA also use an English lexicon whose entries include zones covering syntactic information, a semantic structure (typically linking the entry to the ontology), and calls to procedural semantic routines (e.g., to provide for the reference resolution of pronouns and other deictics). The memories of all the virtual agents are modeled using fact repositories, which are collections of assertions formulated as indexed instances of ontological concepts: e.g., SWALLOW-*78. All knowledge in the MVP environment is recorded using the metalanguage of Ontological Semantics (Nirenburg and Raskin 2004). For lack of space, this metalanguage will not be described here. For the purposes of this paper, suffice it to say that the metalanguage is frame-based, and ontological concepts and their instances are modeled as named sets of property-value pairs. At this time, we are not modeling miscommunication among agents; therefore, in a simplification of the real-world state of affairs, the ontologies of different agents in the MVP do not contain contradictions but, rather, differ only in the coverage and grain size of world description.

MVP will serve as a concrete substrate for our discussion of two kinds of paraphrase that must be treated in modeling embodied cognitive agents.

1. PhysA → CogA Paraphrase involves the conversion of physiological stimuli, like symptoms, from the medically sophisticated form used by PhysA into a lay form that can be interpreted and remembered by CogA.

2. Natural language (NL) → CogA Memory Paraphrase involves two stages: (a) creating a formal meaning representation from NL input (e.g., a question), and (b) matching that representation with those elements of CogA memory that can be used to respond (e.g., the remembered facts needed to answer a question).

We will orient the discussion around a specific example: the doctor in training who is using MVP must find out if

[1] Patent pending.

the virtual patient suffers from the symptom *dysphagia*, which is difficulty swallowing that might involve food and/or liquid moving more slowly than usual through, or getting stuck in, the esophagus. (Note that we concentrate on dysphasia just for purposes of exposition: any other symptom will require the same kinds of paraphrase processing.) For the dysphasia example, our two types of paraphrasing are relevant as follows:

1a. PhysA → CogA Paraphrase. Whereas PhysA tracks the INTENSITY of the concept DYSPHAGIA over time in the simulation, CogA's ontology certainly does not have this concept; moreover, CogA might not even recognize this as a differentiated symptom – it might be perceived, for example, as a vague discomfort after swallowing. Therefore, when PhysA generates a symptom, CogA must interpret it using ontological primitives that it understands. The translation from the PhysA ontological representation to the CogA ontological representation can yield different paraphrases for different patients, though we expect each patient to use just one paraphrase consistently when remembering a given type of symptom.

2a. NL → CogA Memory Paraphrase. The CogA of most patients will not know the English word *dysphagia*, meaning that discussion of this notion with a physician must at least initially use other terms that are known to CogA. If the physician happens to use a formulation that does not precisely match the facts stored in CogA's memory, CogA must be able to recover and make the conceptual link using paraphrase processing.

In the next section, we describe how the CogA in our system paraphrases the inputs from the PhysA and represents them in terms of its own ontology. Following that, we describe how CogA takes basic meaning representations of incoming dialog turns (obtained through its basic language understander, the OntoSem analyzer; see Beale et al. 2003) and connects the events or states mentioned there with events or states in its memory (fact repository), even if the latter are expressed as (conceptual) paraphrases of the mentioned inputs. Figure 1 illustrates the processes we discuss in the paper.

Creating Memories of Life Experiences

Virtual patients (VPs) in MVP "live" over time, their pathophysiology changing according to disease scripts that are parameterized to give each patient a particular, differentiated manifestation.[2] As VPs live, they experience symptoms, make lifestyle choices, and have various medical and non-medical events happen in their lives. The path of their disease and treatment centrally depends upon their own decisions and the decisions of the human attending physician (the user of MVP). The perception of such events and their management in the agent's memory must be modeled in a realistic way, meaning in a way that is compatible with results of earlier perception episodes stored in CogA's memory.[3]

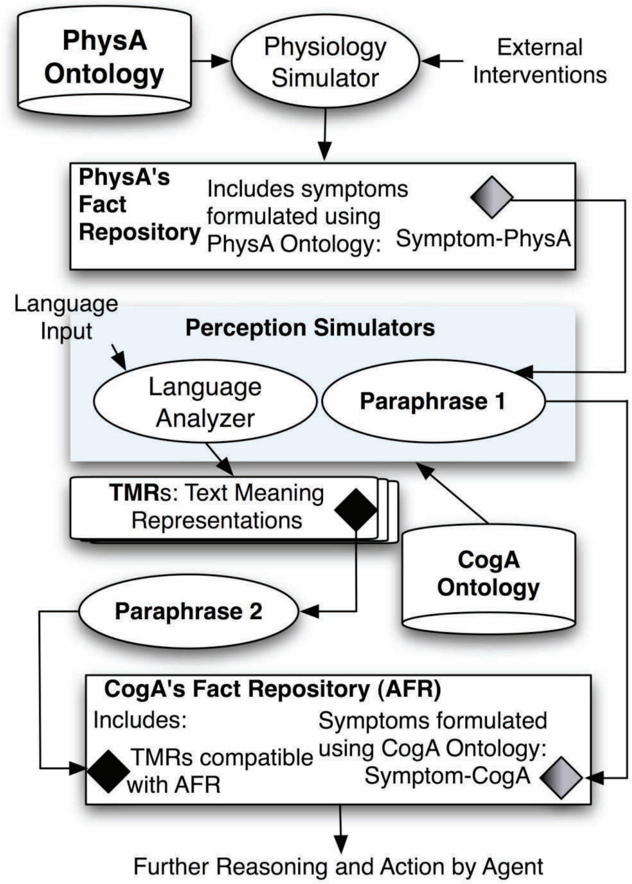

Figure 1. A schematic view of the two kinds of paraphrases in modeling agent perception.

As described earlier, PhysA outputs symptoms, like the INTENSITY of DYSPHAGIA, that might not be directly "understood" by CogA. So the question is, how can one model the communication between PhysA and CogA when they use different ontologies? To explain this, we must describe more closely the ontologies used by PhysA and CogA.

The simulation engine (PhysA) uses an ontology that reflects the scientific community's knowledge of physiology and pathophysiology. It contains detailed scripts describing normal and abnormal physiology, diagnosis and treatment practices, and so on. For example, the SWALLOW script includes over a hundred subevents including the firing of various nerves, the contraction and relaxation of various muscles, and the stepwise movement of swallowed matter though conceptually divided segments

[2] The MVP environment supports the authoring of a library of virtual patients that show clinically relevant distinguishing features.

[3] The modeling of memory management in the MVP addresses such important issues as remembering salient things for an appropriate length of time (in other words, modeling forgetting) and generalizing over repeating states and events.

of the esophagus. The grain size of description of these events is dictated by the needs of simulation and clinical capabilities: since certain drugs can affect nerve firing and therefore affect swallowing, nerve firing must be explicitly included in the swallow script; as such, if the patient takes the given drugs, its swallowing will be automatically affected accordingly.

The PhysA ontology also contains descriptions of symptoms, including DYSPHAGIA; we show only a small subset of its properties below. Note that we are not including asterisks before the numbered ontological instances in order to make the descriptions more readable):

```
(DYSPHAGIA
  (IS-A ANIMAL-SYMPTOM)
  (EXPERIENCER ({ACHALASIA/GERD/LERD}-PATIENT ))
  (SYMPTOM-OF-DISEASE GERD LERD ACHALASIA)
  (PRECONDITION
    (or
      (WIDTH
          (DOMAIN (or LUMEN-{50,58,59,60}))
           (RANGE (< 1.8)))
      (PRESSURE (DOMAIN LES) (RANGE (> 33)))
      (PERISTALSIS-EFFICACY (< .9)))))
  (CAUSED-BY
    (or MOTION-EVENT-{50,58,59,60,70,78,79,80}))
```

The IS-A, EXPERIENCER and SYMPTOM-OF-DISEASE slots should be self-explanatory. The first of the three disjunctive preconditions giving rise to DYSPHAGIA is that the LUMEN (hollow cavity) of certain ESOPHAGEAL-SEGMENTs into which a swallowed substance – called a BOLUS – tries to pass is too narrow (less then 1.8 cm) – which can happen, for example, in the presence of a tumor. The second precondition is that the lower esophageal sphincter (LES, the sphincter between the ESOPHAGUS and the STOMACH), is too tight to permit the BOLUS to readily pass through. The third precondition is that there are insufficient peristaltic contractions pushing the BOLUS down the ESOPHAGUS. Any of these eventualities can only occur if a person attempts to swallow. The effect (not shown above) of any of these preconditions being true is the posting of a specific symptom for a specific location in the esophagus.

Consider the following example, referring to the excerpt of MOTION-EVENT-59 below while doing so. (Note that many of its preconditions and effects are not shown for reasons of space.) Assume that LUMEN-59[4] has a WIDTH of 1.5 cm, and the VP tries to swallow a SOLID-BOLUS. MOTION-EVENT-59, which should take the bolus from LUMEN-58 to LUMEN-59, will (a) occur more slowly than normal (taking 1 second instead of .2 seconds) and (b) an instance of the symptom DYSPHAGIA with an INTENSITY of 2 (on a scale of 0-4) and a LOCATION of LUMEN-59 will be output by the simulator that runs PhysA. This symptom instance will then be passed to CogA.[5]

```
(MOTION-EVENT-59
  (tracks
    (if
      (WIDTH
          (DOMAIN (or LUMEN-{50,58,59,60}))
           (RANGE 1.5 (MEASURED-IN CM)))
    then
      (and
        (DYSPHAGIA
          (INTENSITY 2)
          (LOCATION LUMEN-59))
        (MOTION-EVENT-59.DURATION
          (DOMAIN MOTION-EVENT-59)
          (RANGE 1 (MEASURED-IN SECOND))))
  (PRECONDITION (LOCATION-15))
  (THEME (SEM SOLID-BOLUS-1))
  (SOURCE (SEM LUMEN-58))
  (DESTINATION (SEM LUMEN-59))
  (DURATION (DEFAULT .2) (MEASURED-IN SECOND))
  (EFFECT (LOCATION-16))
 (LOCATION-15
  (DOMAIN (SEM SOLID-BOLUS-1))
  (RANGE (SEM LUMEN-58)))
(BOLUS-1
  (THEME-OF (SEM SWALLOW [+ ontological instance
number assigned to the original SWALLOW event])))
(LOCATION-16
  (DOMAIN (SEM SOLID-BOLUS-1))
  (RANGE (SEM LUMEN-59)))
```

Let us now compare PhysA's understanding of swallowing and dysphagia to that of a typical patient. The SWALLOW script in the CogA ontology might contain as few as two subevents: the volitional action of gulping the food/drink, and the automatic action of it somehow moving downward, ending up in the stomach. It is highly unlikely that a patient would imagine swallowing as having a long series of MOTION-EVENTs that are part of a larger event of PERISTALSIS and can be impeded by specific kinds of physiological malfunctions. As a result, CogA will have to express results of perception episodes associated with swallowing in its memory using instances of the limited related ontological knowledge about swallowing available to it.

An instance of the symptom called DYSPHAGIA in the PhysA ontology might be represented in many different ways in the CogA memory. The following are just a few of many realistic paraphrases. The structures are ontological instances of concepts in the CogA ontology, as stored in the CogA memory. For readability, the examples are presented in a simplified format, with only a relevant

[4] To reiterate, numbered concepts in scripts are *ontological instances*: e.g., many different LUMENs are referred to in the SWALLOW script, each bound to a particular ESOPHAGEAL-SEGMENT.

[5] Remember that the entire SWALLOW script contains hundreds of subevents such as the above; each of these subevents can post symptoms that the cognitive agent must perceive and interpret.

subset of their properties shown. The same ontological instance numbers (which would be preceded by * if we were not simplifying the structures), illustrate co-reference. The rest of the indices signify ontological instances as well. English glosses are provided **only** for orientation.[6] Note that (1a) would be appropriate only if the VP had a background in clinical medicine or had been taught about its disease by the attending physician. (1d) is further explained in Section 3.

1a. dysphagia:
(DYSPHAGIA (INTENSITY (> 0)))

1b. discomfort in one's esophagus when swallowing:
(DISCOMFORT-1
(INTENSITY (> 0)) (LOCATION ESOPHAGUS-*1)
(CAUSED-BY SWALLOW-*1))
(SWALLOW-*1
(AGENT HUMAN-*1))
(ESOPHAGUS-*1
(PART-OF-OBJECT HUMAN-*1))

1c. food gets stuck when swallowing:
(CHANGE-EVENT-*1
(CAUSED-BY SWALLOW-*1)
(PRECONDITION MOTION-EVENT-*1)
(EFFECT MODALITY-*1))
(SWALLOW-*1
(AGENT HUMAN-*1) (THEME FOOD-*1))
(MOTION-EVENT-*1
(THEME FOOD-*1))
(MOTION-EVENT-*2
(THEME FOOD-*1)
(PRECONDITION MOTION-EVENT-*1))
(MODALITY-*1
(TYPE EPITEUCTIC) (VALUE 0)
(SCOPE MOTION-EVENT-2))

There are several ways in which the paraphrasing of interoception could be implemented, ranging from *ad hoc* to sophisticated. The *ad hoc* way is to create an inventory of symptom paraphrases like the one above and then assign one paraphrase to each virtual patient as the way it will remember each symptom. This assignment could be done during patient creation (when physiological features, personality traits, etc., are selected) or automatically – either randomly or heuristically. As regards the latter, a level of medical sophistication could be assigned to each paraphrase and matched to the global level of medical sophistication assigned to each patient. Considering that each disease involves a relatively small number of symptoms, this methodology would not be unduly labor-intensive. The result would be a realistic degree of variation in how patients conceptualize and talk about their symptoms.

However, there are theoretically more interesting, generalizable ways to handle the problem of paraphrasing between PhysA and CogA. We have developed a paraphrase algorithm to convert the medically sophisticated representations generated by the PhysA into the less specialized representations that might be expected of "lay" patients. The algorithm takes as input a structure from PhysA's memory, consisting of instances of concepts in the PhysA ontology, O_{PHYS}, and generates a paraphrase structure that consists of instances of concepts in CogA's ontology, O_{COG}. As is clear from the discussion above, paraphrases generated by this algorithm are not fully meaning-preserving: in the general case, the generated paraphrase describes the same event at a coarser-grain level of description.

PhysA-to-CogA Paraphrase Algorithm

This algorithm takes a message describing a symptom generated by PhysA and simulates interoceptive perception by generating the representation of the symptom in terms of CogA's ontology. This representation is then stored in CogA's memory. Our initial version of the algorithm ("Version 0") uses a relatively small set of heuristics. At the time of writing, the algorithm is being tested and evaluated. We do expect to introduce modifications and additional, finer-grain heuristics into it. At present, the algorithm checks whether concepts whose instances are present in the input structure are shared by O_{PHYS} and O_{COG} and if so, moves these instances, together with their property-value pairs, to the output without modification. We make a simplifying assumption, reasonable in our simulation and modeling environment, that if the concepts are present in the ontologies of more than one agent, these concepts are identical; that is, we do not permit semantic shifts across ontologies.

If the input contains an instance of a concept, $I_{MISSING}$, that is missing from O_{COG}, then we look for a concept, $C_{CLOSEST}$, that appears in both ontologies such that, according to some measure, it is closest to the O_{PHYS} concept missing from O_{COG}. If such a concept is found, then we instantiate it as $I_{RECONSTRUCTED}$, in the output and add to it any of the properties of $I_{MISSING}$ that are defined in $C_{CLOSEST}$. (The properties of $I_{MISSING}$ that are not defined in $C_{CLOSEST}$ are simply omitted.) If the filler of a property passed from $I_{MISSING}$ into $I_{RECONSTRUCTED}$ is a concept or a concept instance, then we make sure that it exists in O_C. If it does not, we apply to it the same procedure as the one we just described for finding the closest substitute.

PhysA-to-CogA Paraphrase Algorithm, Version 0

Let C(I) denote a concept of which I is an instance.
Let $\{I_{PHYS}\}$ denote a set of "top-level" instances of ontological concepts from PhysA's ontology that are present in the input message from PhysA to CogA. An

[6] Note that the language strings provided as glosses can be automatically generated by our system from their corresponding formal representations and vice versa. That capability comes into play when modeling communication between artificial agents and humans.

instance is top-level if it heads a frame in an interoceptive input message from PhysA to CogA. The result of CogA's perception of PhysA's interoceptive message, which will be stored in CogA's memory, is a set of instances of ontological concepts from CogA's ontology. Let $\{I_{COG}\}$ denote the set top-level instances of this output structure.

foreach top-level instance I_P^I in the input,
if $C(I) \in O_C$
then { add this instance as I_C^J to output structure;
Insert-Properties(I_P^I, I_C^J) }
else add Create-Closest-Paraphrase(I_P^I) to output structure.

procedure Insert-Properties(I_{INPUT}, I_{OUTPUT})
foreach property P in I_{INPUT}
if P is defined in $C(I_{INPUT}) \in O_{COG}$
then if Filler F of P is a concept $C \in O_{PHYS}$ or a concept instance I such that $C(I) \in O_{PHYS}$ and $C \notin O_{COG}$, $C(I) \notin O_{COG}$
then insert Create-Closest-Paraphrase(F) as filler of property P in I_{OUTPUT}
else insert property P and its value from I_{INPUT} in I_{OUTPUT}.

function Create-Closest-Paraphrase(I)
if C(I) is an ontological instance
then if the concept of which it is an instance is in O_{COG}
then **return** this concept
else if subsumption hierarchies of O_{PHYS} and O_{COG} up to ancestors with rank below 5 in the ontology[7] share elements
then return the lowest, closest common ancestor
else if C(I) has a value set for any of the properties PART-OF-OBJECT or PART-OF-EVENT, and a subset of elements of this value set is a concept is in O_{COG}
then return any member of this subset
else if C(I) has a value set for any of the properties PRECONDITION or EFFECT, and a subset of elements of this value set is a concept is in O_{COG}
then return any member of this subset

The paraphrase algorithm is triggered each time PhysA posts a SYMPTOM to be interoceptively perceived by CogA.

Let us illustrate its operation on a sample symptom:

```
(DYSPHAGIA-12
    (INTENSITY .2)
    (LOCATION     LUMEN-59)
    (CAUSED-BY    SWALLOW-36))
```

[7] This constraint serves to exclude concepts that belong to what is known as the "upper" ontology, not a domain ontology.

In this section we will look at how the paraphrase algorithm can generate the paraphrase shown in (1b); it can also generate the paraphrase in (1c) but space does not permit us to detail this process.

Most CogA ontologies will not contain the concept DYSPHAGIA but will contain its ancestor concept DISCOMFORT.[8] (The path is DYSPHAGIA < DISCOMFORT < SYMPTOM < …) So, DYSPHAGIA can be replaced by DISCOMFORT. The properties INTENSITY, LOCATION and CAUSED-BY are defined in all the ontologies in the MVP environment, as is the event (script) SWALLOW. CogA will likely not understand LUMEN-59: first, it may not know what a LUMEN is to begin with; second, it will not be able to interpret a (numbered) ontological instance of a concept that belongs to a highly specified version of the SWALLOW that is missing from its ontology. However, the swallow script in PhysA's ontology contains information that can be used to paraphrase LUMEN-59:

```
(LUMEN-59
   (ONTOLOGICAL-INSTANCE-OF LUMEN)
   (PART-OF-OBJECT ESOPHAGEAL-SEGMENT-59))
(ESOPHAGEAL-SEGMENT-59
   (PART-OF-OBJECT ESOPHAGUS))
```

Using the meronymic property PART-OF-OBJECT, the paraphrase algorithm traces a path from a concept the CogA does not know – LUMEN-59 – to a concept it does know – ESOPHAGUS. Note that even if the CogA does not know the word 'esophagus', it will still have the concept of esophagus in its ontology, since most people know that there is a body part that connects the mouth to the stomach. We have chosen to call that body part ESOPHAGUS in the ontologies of all of our patients. The paraphraser algorithm, thus, can output the following structure, to be stored in CogA's memory, which matches the knowledge structure in (1b).

```
(DISCOMFORT -23
    (INTENSITY .2)
    (LOCATION  ESOPHAGUS-1)
    (CAUSED-BY   SWALLOW-36))
```

Once symptoms have been paraphrased using the expressive means available in the CogA ontology and have been stored in CogA's memory, CogA can refer to them when making decisions about its health care and when communicating with a physician.

[8] All of our patients' ontologies contain general medical concepts like SYMPTOM, DISCOMFORT, DISEASE and MEDICATION, as well as a large inventory of OBJECTS, EVENTS and PROPERTIES that are part of "universal" world knowledge.

Understanding Natural Language

In addition to interoception, the MVP's cognitive agent can perceive the world through understanding natural language.

Each agent – real or virtual – has an idiosyncratic lexicon, ontology and fact repository (memory); as such, people adjust their utterances to their expectations of their interlocutor's knowledge. For example, rather than ask a patient, "Do you have dysphagia?", a physician would more naturally ask, "Does food ever get stuck when you swallow?"

When the virtual patient, or any of the agents in any applications of Ontological Semantics, receives language input, it processes it using its personal ontology, lexicon and memory, in conjunction with the OntoSem semantic analyzer. The goals of processing are: a) to disambiguate the input, b) to carry out all necessary reasoning (including reasoning for reference resolution and the interpretation of indirect speech acts), and c) to store the results of analysis in its memory using meaning representations like the ones shown above. As we mentioned earlier, the glosses of examples (1a)-(1c), if used as input to the OntoSem analyzer, would produce the associated formal structures. The core process of semantic analysis essentially compares the selectional constraints of heads with their arguments and adjuncts and selects the most felicitous combination of senses (Beale et al. 2004).

For reasons of space, we cannot provide a complete explanation of how these meaning representations are generated, but we can provide an example that will convey the gist. Consider the input *Food gets stuck when I swallow* (cf. (1c) above). The OntoSem lexical entry for this meaning of stick is shown below.

```
(stick-v1
   (cat v)
   (ex "The wheels (got) stuck (in the mud).")
   (syn-struc
      ((subject ((root $var1)(cat np)) (root $var0)(cat v))
   (sem-struc
      (CHANGE-EVENT
         (PRECONDITION (value refsem1))
         (EFFECT (value refsem3)))
      (refsem1
         (MOTION-EVENT (THEME (value ^$var1))))
      (refsem2
         (MOTION-EVENT
            (PRECONDITION (value refsem1))
            (THEME (value ^$var1))))
      (refsem3
         (MODALITY
            (TYPE epistemic) (VALUE 0)
            (SCOPE (value refsem2))
            (ATTRIBUTED-TO (or ^$var1 *speaker))))
      (MODALITY
         (TYPE evaluative) (VALUE < 0.3)
         (SCOPE (value ^$var0)) (ATTRIBUTED-TO *speaker)))))
```

The syntactic structure (syn-struc) indicates that it is an intransitive sense. The semantic structure (sem-struc) says that this is a CHANGE-EVENT – an EVENT whose meaning is defined in terms of its preconditions and effects.[9] The refsem notation introduces reification of relation fillers. The caret, ^, means "the meaning of," so that, for instance, ^$var1 means "the meaning of whatever the value of the syntactic variable $var1 is." This is a device for encoding linking, which is a syntax-to-semantics mapping. The precondition for this sense of *stick* is a successful MOTION-EVENT with a certain THEME, and the EFFECT is an unsuccessful MOTION-EVENT with the same THEME. The assignment of epistemic modality with a value of 0 scoping over the second MOTION-EVENT indicates its failure. The assignment of the evaluative modality reflects the meaning that getting stuck is undesirable for the agent of the action and the speaker. So, instead of pointing to a concept STICK in the ontology, this meaning is explained using preconditions and effects: sticking can only happen if something has been in motion and is now not in motion, and the lack of motion is an undesirable thing. This sem-struc is used in the lexicon entries for the appropriate senses of all of the synonyms of *getting stuck*, such as *getting caught, catching up, sticking,* etc.

Searching Memory for Matching Events/States

As stated earlier, the main job of agent to agent paraphrase is to create memories. The matching of new memories to old memories must be done as part of memory management, but this is unproblematic: the memories of a given type of event will always be recorded the same way for a given agent. However, when it comes to language processing and the expected communication between artificial agents and a human, matching becomes a big issue.[10] In this section we talk about CogA's task of matching questions posed by a person to its own stored memories so that it can adequately answer the question.

Below is a sampling of sources of mismatch between physician input and CogA memory, and the means we use to overcome them.

A. String-level paraphrase. OntoSem text processing can reduce the meaning of essentially synonymous strings to a single formal meaning representation. For example, lexical synonyms (be they words or phrases) and different diathesis transformations (e.g., active and passive voice) yield the same meaning representation. Likewise, a single representation is produced for inputs with semantic inversions – *X led to Y* will produce the same representation *as a result of X, Y*. Another widespread source of paraphrase that is resolved by basic semantic

[9] CHANGE-EVENTs are a staple of conceptual representation; cf. Minsky's (2006) Trans-Frames (p.283) and McShane 2008 (forthcoming).

[10] Creating new memories based on language input is also done, but it is outside the scope of this paper.

analysis is the use of light verbs: e.g., OntoSem will produce the same output for *carry out an inspection* and *inspect*.

B. Reference resolution. Reference resolution is needed for all kinds of semantically oriented text processing. Our approach to reference resolution is to link all referring expressions to their fact repository *anchors*, using textual coreference as a heuristic but not as a stopping point for reference resolution. We are currently working on improving our reference resolution capabilities so that if, for example, the physician's input is *Does it hurt?*, CogA can resolve *it* to the correct real-world anchor in its memory, and can try to match a fully anchored knowledge structure from the text input to the relevant structures in its memory.

C. Specificity/vagueness. One can talk about objects and events using very different levels of specificity. For example, a physician might be asking about *discomfort* in the *esophagus* whereas the VP has a memory of some *symptom* in its *chest*.

D. Level of detail. The VP might remember that it experienced discomfort when it swallowed, with no location specified, whereas the physician might ask about discomfort in the esophagus when swallowing. Similarly, the physician might ask about the difference in symptoms between swallowing food and drink, whereas the VP never thought to remember them separately – he thinks of all ingestion as the same.

E. Term vs. explanation. A common locus of paraphrase involves using a known term versus an explanation of that term. E.g., the physician might refer to the *esophagus* whereas the patient might think of the esophagus as *the body part between the mouth and the esophagus*: (ANIMAL-PART (DISTAL-TO MOUTH) (PROXIMAL-TO STOMACH)). Or, the physician might use the term *bolus*, which the patient does not know, and have to follow up with an explanation of what a bolus is – swallowed food or drink.

F. Wordy vs. pithy. Some people are very wordy whereas others are pithy. Functionally superfluous wordiness must be stripped from the meaning of text input in order for CogA to match it with stored memories. E.g., the physician might ask, *Let's say you're eating dinner and you go to swallow some food, or even drink, and while you are swallowing something feels not quite right – does that ever happen to you?* Or he could just say *Do you have trouble swallowing?*

This short non-exhaustive inventory of paraphrase issues is sufficient to convey the nature and scope of problems associated with having an artificial agent **interpret** new input, **link** that input to stored memories, and **leverage** the relevant stored memories to carry out cognitive tasks, like answering questions. The abovementioned heuristics developed for the paraphrasing algorithm between sets of ontological concept instances can be reused for the treatment of linguistically perceived paraphrases. This is made possible because after the language input is analyzed, it is represented in exactly the same way as are results of interoception. However, additional heuristics are needed for the just mentioned issue F (although we expect wordiness from some people, we do not expect it of intelligent agents) and for some subtypes of issues C-E.

Discussion

Recently advanced dialog systems have been developed and deployed in the medical domain (e.g., Allen et al. 2006). We believe that developing cognitive models of patients based on simulated physiological and pathological processes incorporates that objective while adding a cognitive modeling angle to it.

Lexical and ontological lacunae call for a useful application of machine learning. Ideally, when faced with a knowledge gap (like lack of knowledge of the terms *dysphagia* and *bolus*), a system will take the initiative by asking a clarification question, then understand the human's response as a definitional descriptive paraphrase and automatically create a lexicon entry or an ontological concept that will help in treating the original input and will enhance the agent's knowledge in general. We are currently developing such machine learning capabilities. Within the MVP project, we are enabling agents to learn by being told: e.g., the human doctor explains to the patient that his difficulty swallowing is actually called *dysphagia,* and the patient stores that knowledge as a memory in his ontology and lexicon. Outside of MVP, we are working on automating the acquisition of lexicon and ontology by exploiting our system's ability to semantically analyze text. This approach, which we call "learning by reading by learning to read," promises to ease the knowledge bottleneck and thereby make the methods that we develop for smaller domains more readily applicable to wider domains (Nirenburg and Oates 2007).

In conclusion, this paper has described an approach to modeling the augmentation and management of an intelligent agent's memory as a result of the latter's perception and interoception. A characteristic feature of this approach is the use of the same ontological metalanguage and the same structure of the agent's memory to store the results of different perception - interoception systems. Indeed, the same knowledge resources are used for encoding both symptoms obtained from the simulated physiological agent and the meaning of natural language messages that the cognitive agent receives from the human during a dialog. At the moment, the system addresses only two kinds of perception – interoception and language. We believe that this modeling, based as it is on simulation and paraphrase and incorporated into an ontologically-grounded memory, can be applied also for other kinds of input, for example, tactile and visual.

Finally, our approach to paraphrasing has promise in more traditional application areas of NLP, such as machine translation. If one can generate a set of meaning-preserving paraphrases for an input text, then the probability that at least one of these paraphrases will lend itself to a goodquality translation into a target language will be enhanced. This is true irrespective of which specific approach to machine translation is used.

References

Allen, J., Ferguson, G., Blaylock, N., Byron, D., Chambers, N., Dzikovska, M., Galescu, L, and Swift, M. 2006. Chester: Towards a Personal Medication Advisor. *Journal of Biomedical Informatics*, 500-513.

Beale, S., Nirenburg, S., and McShane, M. 2003. Just-in-time Grammar. In *Proceedings of the 2003 International Multi-conference in Computer Science and Computer Engineering*. Las Vegas, Nevada.

Beale, S., Lavoie, G., McShane, M., Nirenburg, S., and Korelsky, T. 2004. Question answering Using Ontological Semantics. In Proceedings of the ACL-2004 Workshop on Text Meaning and Interpretation. Barcelona, Spain.

McShane, M. 2008. Achieving human-level NLP: The case of reference resolution. In preparation.

McShane, M., Nirenburg, S.,. Beale, S., Jarrell, B., Fantry, G. 2007. Knowledge-based Modeling and Simulation of Diseases with Highly Differentiated Clinical Manifestations. In Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 07). Amsterdam, The Netherlands.

McShane, M., Nirenburg, S., and Beale, S. 2008. Ontology, Lexicon and Fact Repository as Leveraged to Interpret Events of Change. *Ontologies and the Lexicon*, Cambridge University Press Studies on Natural Language Processing. Forthcoming.

Minsky, M. 2006. *The Emotion Machine*. New York, NY: Simon & Schuster.

Nirenburg, S., and Oates, T. 2007. Learning by Reading by Learning to Read. In Proceedings of the International Conference on Semantic Computing (ICSC-07). Irvine, CA.

Nirenburg, S., and Raskin, V. 2004. *Ontological Semantics*. Cambridge, MA: MIT Press.

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures.* New York: L. Erlbaum Associates.