Proceedings of the Twelfth Annual Conference on Advances in Cognitive Systems ACS-2025 (369-388)

Pursuing Actionable Perception Interpretation in Cognitive Robotic Systems

Marjorie McShane MARGEMC34@GMAIL.COM
Sergei Nirenburg ZAVEDOMO@GMAIL.COM
Jesse English DRJESSEENGLISH@GMAIL.COM
Sanjay Oruganti SANJAYOVS.RPI@OUTLOOK.COM

Language-Endowed Intelligent Agents Lab, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract

Semantically interpreting and grounding multimodal stimuli is a core requirement of cognitive robotic systems, but it is challenging because inputs can be fragmented, ambiguous, underspecified, ill-formed, and conveyed through noisy channels. This means that agents, like people, need to be able to determine when their understanding is *actionable*—i.e., sufficient to support reasoning about action—even if it is imperfect or incomplete. When an interpretation is not actionable, the agent has to decide what to do, such as wait and see what happens or seek clarification through dialog. This paper demonstrates that it is possible to model actionability assessment, as well as recovery from non-actionable interpretations, without drowning in real-world complexity by modeling agents as *collaborative social* agents. Like human apprentices, such agents can take best guesses in benign contexts, ask clarification questions, and generally rely on their human partners to share the responsibility for achieving a successful collaboration. The paper also briefly comments on another use of the term *actionability*, which involves the agent's ability to actually carry out an action that it understands it should do. The models reported in the paper are implemented in Language-Endowed Intelligent Agents configured within the HARMONIC neurosymbolic architecture.

1. Introduction

When cognitive-robotic agents are assisting people in real-world tasks, they are receiving streams of multimodal stimuli that can involve language, vision, haptics, and other sensor inputs. Those inputs can be difficult to interpret since they can be fragmented, ambiguous, underspecified, ill-formed, incomplete, and conveyed through noisy channels. This means that agents, like people, need to be able to determine when their understanding is *actionable*—i.e., sufficient to support reasoning about action—even if it is imperfect or incomplete. When an interpretation is not actionable, the agent has to decide what to do, such as wait and see what happens or seek clarification through dialog.

For example, if two people and a cognitive robot are collaborating on an engine repair task and one of the people says *We need a clamp,* this could be a request for the agent to fetch a clamp, a request for the other human to fetch a clamp, an explanation of the speaker's current action (e.g., walking away to get a clamp), or something else entirely (maybe the team has no clamps so they have to come up with a plan B). Selecting the intended interpretation requires reasoning about the situation overall and everyone's role in it. The model of actionability assessment presented below

This work is licensed under a Creative Commons Attribution International 4.0 License.

enables agents to handle a large number of such eventualities in ways that we think their human partners will find useful.

This paper makes three main claims:

1. Since it is beyond the state of the art for cognitive robots to fully and confidently interpret all multimodal stimuli in complex environments, agent systems must be designed to judge when their interpretation of inputs is actionable and respond accordingly, including by pursuing clarification when needed.
2. Pursuing an actionable interpretation always requires reasoning about action and sometimes involves taking action as well, as by asking a question and then interpreting the response. This means that cognitive architectures that place perception, reasoning, and action in a pipeline are too simplistic to accommodate human-like behavior.
3. It is possible to model actionability assessment and recovery from non-actionable interpretations without drowning in real-world complexity by modeling agents as *collaborative social* agents. Like human apprentices, such agents can take best guesses in benign contexts, ask clarification questions, and generally rely on their human partners to share the responsibility for achieving a successful collaboration.

Evidence for these claims includes the real-world and linguistic phenomena that make the actionability model necessary, the model itself, and examples implemented within the framework of Language-Endowed Intelligent Agents (LEIAs). LEIA research is detailed in two recent open-access books: *Linguistics for the Age of AI* (2021) and *Agents in the Long Game of AI* (2024), which will be referred to hereafter as LingAI and LongGame, respectively. As regards the new contribution of this paper, although actionability is referred to in these books, the model presented here was not worked out in detail until recently, and this is the first time it is being reported.

The title of the paper specifies that we are focusing on "actionable perception interpretation". This intersects with but is not identical to another use of the term actionability, which involves whether an agent can successfully carry out a task. Assessing whether an action can be carried out can require delving deeply into planning, mental simulation, and even assessment by trial and error. This goes far beyond understanding communication, which is the scope of *perception interpretation.*

The answer to modeling the duality of *actionability* is as follows. The agents that our team develops, like people, know about their general capabilities, and they use this knowledge when they are interpreting inputs. However, they do not carry out a full situational assessment of whether they are likely to succeed at a given task in a given situation during input interpretation. To take a human example, if one able-bodied family member says to another able-bodied family member, "I need help moving this desk into the other room," the latter will interpret it as a request for help: moving around small furniture is just not very physically demanding. Arriving at this interpretation doesn't require creating a plan (measuring the desk and the doorway, thinking about the flooring, etc.) and assessing its likelihood for success. We are designing LEIAs to behave similarly. They have knowledge about their general capabilities, which they use during input interpretation. After they have interpreted inputs, they move on to reasoning about action, at which point any foreseen impediments to carrying out the needed action are addressed by a different model.

It must be emphasized that ability-based heuristics are usually not needed when interpreting inputs. They are only needed for interpreting inputs that are ambiguous or underspecified things, such as "I need help with X" rather than the more direct "Help me do X." Sections 2 and 3 below describe the model for assessing the actionability of perception interpretation, while section 4 addresses the second, physically-oriented, use of the term actionability.

1.1 A Brief Introduction to LEIAs

LEIAs are neurosymbolic, multimodal cognitive-robotic systems implemented in the HARMONIC architecture, shown in Fig. 1 (Oruganti et al. 2024a,b). The cognitive (strategic) layer primarily relies on knowledge-based modeling to support reliability, transparency, and explainability. The robotic (tactical) layer primarily relies on machine learning, which is effective and sufficient since the associated capabilities (e.g., the "how" of moving a robotic arm or avoiding collisions) need no explanation. The components of the cognitive and robotic layers function both independently and interactively.

LEIAs represent a novel approach to Agentic AI, which we call OntoAgentic AI. Whereas typical Agentic AI systems use language models both as the orchestrator and for support functions, OntoAgentic AI uses a LEIA as the orchestrator and leverages both LEIA agents and language-model-based systems for support functions. OntoAgentic AI, therefore, offers reliable, explainable control of overall system operation as well as the cognitive operation of each individual LEIA.

LEIA cognition orients around meaning, which is defined in terms of an unambiguous, language-independent ontology, following the theory of Ontological Semantics (Nirenburg & Raskin, 2004). When LEIAs interpret stimuli, reason, or plan, they do it in terms of ontologically-grounded meaning representations.

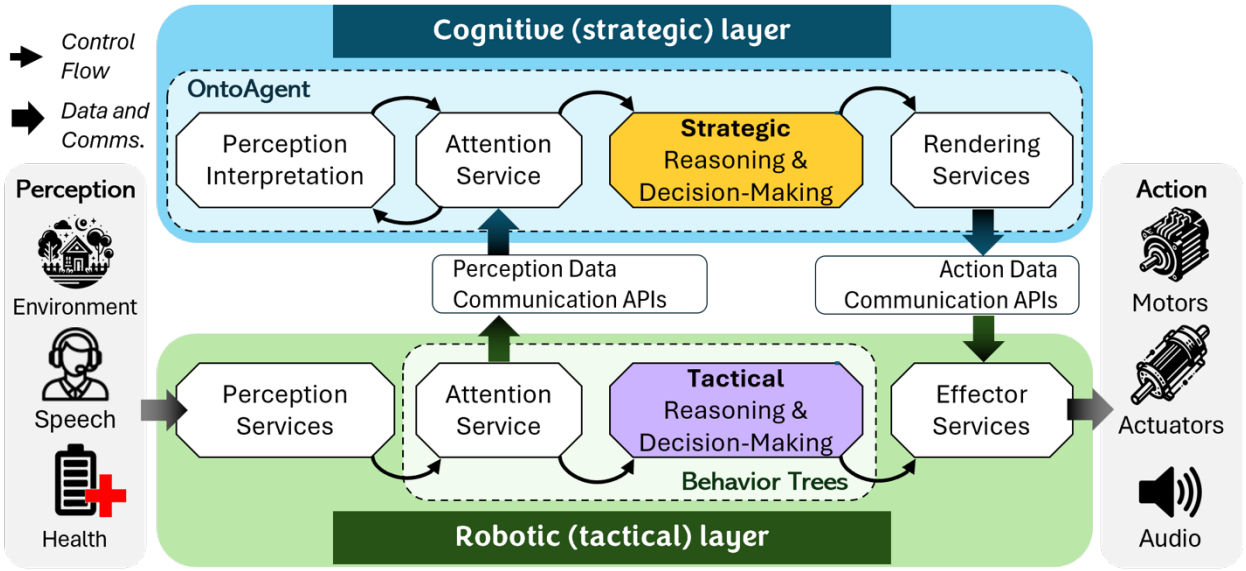

Fig. 1 The HARMONIC cognitive-robotic architecture.

For example, whether an agent sees a red-hot pipe, is told that the pipe is extremely hot, or touches it so that its sensors sense the burn, it will remember the same ontologically-grounded meaning: that this particular instance of PIPE, recorded in episodic memory as PIPE-1, has the highest value on the abstract {0,1} scale of TEMPERATURE. If an actual temperature was known, it would be recorded along with its measuring unit.

TEMPERATURE-1
  DOMAIN PIPE-1
  RANGE 1
  TIME 2025-07-09T11:35

Depending on how the agent perceived this information, the meaning representation would be decorated with different metadata. For example, if someone said, "That pipe is scorching hot", then the frame above would be the THEME of an instance of ASSERT; the AGENT (speaker) and BENEFICIARY (hearer) would be indicated; and the lexical senses that were used to generate the analysis would be indicated.

Preparing agents to interpret perceptive inputs in terms of an ontological metalanguage is difficult and expensive; however, that cost is justified by the gains, which include simplifying reasoning about action, unifying knowledge representation, supporting symbolic cognitive modeling, and allowing the vast majority of agent modeling to be language independent (for additional discussion, see LingAI, section 2.8.1).

2. The Model for Pursuing Actionable Interpretations of Perceptual Stimuli

Fig. 2 shows the top-level model of how LEIAs pursue actionable interpretations of perceptual stimuli. The rows of the algorithm are described in the listed subsections. But before proceeding, we must define what an input is. For purposes of this discussion, it is a whole dialog turn, which might include any number of sentences or fragments along with the non-linguistic stimuli the agent perceives at the time of speech.[1] There are four reasons for processing dialog turns as a whole. First, in task-oriented settings, which are the target of LEIA systems, dialog turns tend to be not very long; and, if they are, then the speaker presumably intends them to be understood at one go. Second, speech happens fast, and agents cannot interpret and respond to inputs fast enough to interrupt—if one would even want that behavior.[2] Third, in order to fully understand a dialog turn, the agent needs to understand the semantic relationships between all of the clauses, fragments, and/or sentences, no matter how they are expressed—which can include highly variable punctuation conventions resulting from speech-to-text processing (cf. section 2.1). Finally, it is common for dialog turns in task-oriented contexts to include one thing that must be acted upon, like a request or a question, along with supporting information. In such cases, the agent doesn't necessarily have to

[1] Details of speech-to-text systems—including how to compute the end of a dialog turn and which punctuation marks (if any) are used—are outside of the scope of this paper.

[2] An exception is reflexive behavior, which can be caused by any type of stimulus: e.g., avoiding collisions using vision or stopping cold when someone yells "Stop!".

understand all of the supporting information in order to respond to the active part, which is a key consideration when reasoning about actionability (cf. section 2.3).

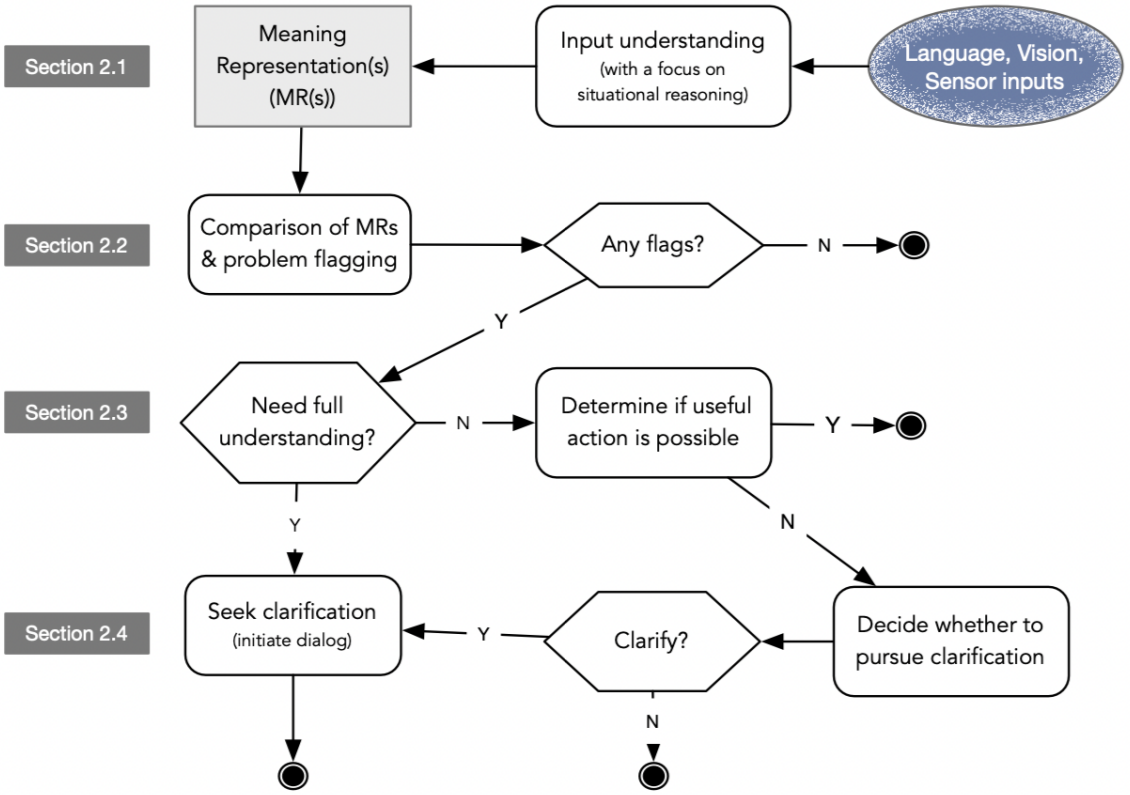

Fig. 2 The top-level model for pursuing actionable perception interpretation.

2.1 Input Understanding with a Focus on Situational Reasoning

When LEIAs are converting perceptual inputs into ontologically-grounded meaning representations, they need to analyze far more than what is overt in the communication. On the one hand, they need to reconstruct elided material and implicatures; on the other hand, they need to understand the purpose of the communication, also known as the communicative act. (*Communicative act* is a more encompassing term than *speech act* since it covers non-linguistic and hybrid communication as well.) For example, when a customer says "Two large lattes" to a barista, the communicative act is placing an order, whereas when the barista repeats this to the cashier, the communicative act is a request to ring up the charge.

Many aspects of ellipsis resolution and inferencing are described in LingAI and LongGame and will not be recounted here. Instead, we limit this discussion to recent progress involving two reasoning-heavy processes at the tail end of the language understanding pipeline that are particularly

important for assessing actionability: (a) reasoning about the semantic relationships between propositions in a dialog turn, particularly when some of them are elided (i.e., not indicated linguistically); and (b) reasoning about the communicative act conveyed by the input, particularly when it is ambiguous, underspecified, or elided. We discuss each of these in turn.

2.1.1 Reasoning about the semantic relationship between propositions in a dialog turn.

Most often, important semantic relationships between propositions are expressed linguistically: e.g., *You need to change the oil* ***because*** *it hasn't been changed in a long time; Check the oil level* ***before*** *you replace the oil cap.* However, it is perfectly normal to not express some such relationships if they are, at least from a human perspective, self-evident.[3] For example, *Open the engine cover, we need to check the oil* implies that the former is the precondition of the latter, and *I'm changing the tire—it went flat this morning* implies that former is caused by the latter.

Past attempts to describe discourse relations have often resulted in unmanageable complexity, as by overcomplicating the definition of *discourse chunk* (i.e., the kinds of entities that need to be connected by discourse relations) and by creating such a large inventory of fine-grained discourse relations that even people could not reliably distinguish between them.[4] In modeling LEIAs, by contrast, we first orient around typical cases that offer useful behavior, and then expand the model's coverage and precision in ways that remain fully computer tractable.

In the current version of the model, we define the discourse chunks of interest as independent clauses, which can be sentences or fragments; we use a relatively constrained inventory of ontologically-defined relations, focusing mostly on temporal, causal, and instrumental ones; and we use straightforward reasoning rules that leverage the agent's ontological knowledge. For example:

- If two propositions have no linguistically indicated relationship but the first can be understood as a precondition for the second, then this relation is inferred: *Open the engine cover, we need to check the oil*.
- If two propositions have no linguistically indicated relationship but the second can be understood as causing the former, then this relation is inferred: *I'm changing the tire, it went flat this morning*.
- If two propositions have no linguistically indicated relationship but the second can be understood as referring to the instrument of the former, then relation is inferred: *We need to open the engine cover. That screwdriver would be good.*

[3] They can be underspecified as well, as by using the vague conjunction *and*, which can imply causality (*The tire went flat and I changed it*), sequence (*He jacked up the car and I removed the tire*) or a vaguer kind of juxtaposition (*We build furniture and repair upholstery*). In such cases, the agent uses similar reasoning as for elided conjunctions. LongGame, section 7.2.2 discusses related issues in the context of automatic learning.

[4] References include the literature and annotations associated with Rhetorical Structure Theory (https://www.sfu.ca/rst/01intro/intro.html; Das & Taboada, 2018); the corpus-annotation effort reported in Carlson, Marcu, & Okurowski (2003); and related psycholinguistic literature, such as Marchal, Scholman, & Demberg (2020). Obviously, an approach that constrains complexity will lose some semantic precision, but that is the very nature of modeling: models are always simplifications of reality (cf. Bailer-Jones, 2009).

- If sequential propositions indicate actions by different people, they are understood to occur simultaneously unless one is a precondition for the other: *I'll paint the wall. You paint the ceiling.*

If the agent's world knowledge is not sufficient for it to infer a specific discourse relation between propositions in a dialog turn, that's fine.[5] On the one hand, it is possible that the relationship is vague and is not needed for reasoning about action. On the other hand, if the lack of a necessary inference leads to the agent behaving in a way the speaker didn't expect, the speaker can take the initiative to straighten things out. For example, if the robot fails to understand the following as an indirect request—*We need to open the engine cover. That screwdriver would be good*—then the person can follow up with an explicit request: *Could you give it to me?* This is perfectly normal human behavior that does not require anything special of the people who will be interacting with LEIAs.

Why do we care so much if the agent is good at inferring the semantic relationships between clauses? Because understanding discourse relations, including elided ones, is important for assessing the actionability of an input interpretation, particularly when agents are permitted to act without complete understanding (cf. section 2.3). For example, if the agent hears, *Work faster, the boss is on the warpath*, it should work faster even if it doesn't understand what *on the warpath* means because there is no reason to believe that the statement about the boss is a reason *not* to work faster. By contrast, if it hears *Open the engine cover. Use the screwdriver poking out of the toolbox,* it shouldn't open the cover unless it understands what *up top* means because that information relates to the instrument of the action and might be important: Maybe the screwdriver located below would strip the screw.

2.1.2 Reasoning About Communicative Acts

What people mean by what they say—i.e., the communicative act—can be overt and precise, overt and underspecified, overt and ambiguous, or elided. As with discourse relations, agents need to zero in on the most realistic interpretation possible to support reasoning about actionability. We will consider each case in turn.

Communicative acts are **overt and precise** when the form of the utterance matches and makes clear the communicative act: e.g., requests and demands use imperative verb forms (*Bring me a hammer*) and questions use question constructions (*Do we have any more brackets?*).

Communicative acts are **overt and underspecified** when the linguistic form of the input makes the communicative act clear but the agent needs a more precise understanding in order to reason about action. For example, yes-no questions can often be detected from their linguistic form (*Did you X? Can he Y? You X-ed, right?,* and so on), so the meaning representation the agent will generate prior to reasoning about the communicative act will include the concept REQUEST-INFO-YN. However, that is too vague for the agent to actually respond to. In fact, it is the root of a much larger ontological subtree, an excerpt of which is as follows:

[5] See Section 6 for the option of consulting a language model for this information.

- REQUEST-INFO
  - REQUEST-INFO-YN
    + REQUEST-INFO-YN-PERCEPTION-RECOG
    + REQUEST-INFO-YN-LEX-ONTO-COVERAGE
      - REQUEST-INFO-ONTO
      - REQUEST-INFO-YN-EPISODIC
      - REQUEST-INFO-YN-AGENDA

The agent detects which subtype of question is being asked by working through detection functions recorded in each of the concepts in the subtree. For example, the detection function in REQUEST-INFO-YN-**EPISODIC** allows the agent to recognize questions about instances of ontological concepts (*Is the engine cool?*), whereas the detection function in REQUEST-INFO-YN-**ONTO** allows the agent to detect questions about generic ontological knowledge (*Can ostriches fly?*). These different communicative acts will play out differently when the agent proceeds to reasoning about action. For example, after detecting a REQUEST-INFO-YN-**EPISODIC**, the agent will instantiate its adjacency pair, RESPOND-TO-REQUEST-INFO-YN-**EPISODIC**, and the procedure recorded therein will guide it in searching its episodic memory for the answer and formulating it appropriately.

Communicative acts are **overt and ambiguous** when a linguistic construction can be interpreted as a direct speech act or an indirect speech act. (The term *speech act,* rather than communicative act, is appropriate here because we are talking specifically about linguistic constructions.) For example, *I need a hammer* can be a request to fetch one or simply a statement that I need one—maybe I know we don't have one or that you can't fetch it. Similarly, *I think we need to get gas* can propose a plan (let's do it) or it can just be a statement of fact—as when it's followed by *but we don't have time because we're already late for our plane, so we'll have to cross our fingers.* All constructions that have indirect-speech-act meanings also have direct-speech-act meanings, and both are recorded the agent's lexicon. This means that every time such a construction is used, the agent will have to resolve the speech-act ambiguity. How do we prepare them to do that?

The first important observation is that the indirect meaning—the one that would require a more active response—tends to be the intended one. So, the simplest model would just have LEIAs select it and occasionally be wrong. Whether or not we do this depends in part on competing development priorities and in part on how well we expect the humans in the loop to tolerate errors by agents. For example, if someone says in frustration, "We need a crane to lift this!" he would not be pleased if the robot interpreted this as a request, tried to create a plan for procuring a crane, failed, and responded by saying, "I don't know how to get a crane." (*Of course you don't, you moronic robot—nobody's asking you to!*)

Since we don't want to unnecessarily aggravate the humans who will interact with our agents, it is worth trying to make the agents at least a little more sophisticated in reasoning about ambiguous speech acts. Consider the case when an agent needs to choose between a request and an assertion, both of which are available interpretations for an utterance like "I'd love a coffee". (The LEIA's lexicon contains two senses of the construction "I'd love a NP": one is a request to provide one, the other is the assertion of a desire—*assertion* being a type of speech act. Note that any assertion of a

desire can be an indirect request for the hearer to try to fulfil it, but the actual action that would need to be taken is highly context dependent.) Below is a simple model of how people likely resolve such ambiguities:

Could I, at least in principle, fulfill the request?
  If yes, then
    Based on social roles, could the person be asking me to this?
      If yes, then I'll select the "request" interpretation. [1]
      If no, then I'll select the "assertion" interpretation. [2]
  If no, then
    Do I believe that the speaker believes that I can fulfill the request?
      If yes, then I'll select the "request" interpretation. [3]
      If no, then I'll select the "assertion" interpretation. [4]

We can see how all four numbered conditions above play out for "I'd love a coffee":

[1] Someone says this to a server in a restaurant: it is a request for a coffee.
[2] Someone says this to her boss during a meeting: it is not a request for a coffee (though, being an expression of a desire, it might be interpreted as a request for something else, like taking a break).
[3] A houseguest says this to you, not knowing that you ran out of coffee: it is a request for a coffee.
[4] A stranger next to you in a plane says this to you: it is not a request for a coffee (though, as above, it might be interpreted in some other way, as a desire to chat or a question about how to get the stewardess's attention).

To implement this reasoning in LEIAs, it makes sense to treat physical and mental actions separately.

LEIAs know their physical capabilities, so if a LEIA knows how to fetch things but not how to drive (see section 4 for how this is recorded in the ontology), it will interpret *I need a hammer* as a request but *I need a lift to the airport* as an assertion. Of course, there can be edge cases for which the LEIA isn't sure: maybe it can fetch things in principle but has never tried to fetch a certain kind of object, so it neither knows whether it could do it nor whether the speaker believes it could. But edge cases do not erase the utility of enabling the LEIA to make quick, straightforward assessments about whether an action-oriented interpretation is, in principle, within its capabilities. If it is, then that is the interpretation it prefers.

For mental actions like answering a question or solving a problem, it is more difficult for the agent to judge whether the speaker might be making a request without actually trying to carry it out the action and seeing if it can. For example, if someone says to a LEIA *We need to figure out how to move this rock*, this could be a request to propose a plan, an explanation for why work is temporarily halted, or an instance of talking to oneself (people talk to themselves all the time and can't be stopped from doing this just because agents are around). To figure out if the person was asking the LEIA for action — i.e., seeking a plan—the LEIA needs to try to create one. If it can, it will prefer that interpretation; if not, it will assume that something else was meant. One can ask, do

people actually do this look-ahead reasoning before settling on an interpretation? We think they likely do, at least by default.

Finally, communicative acts are **elided** in declarative sentences that are not intended to be assertions. For example, if you're building a chair, saying *Now the back leg* proposes starting the next step in the plan. The key to modeling an agent's understanding the propositional semantics and the communicative act of such utterances is for the agent to consult the active plan on its agenda, look at the next anticipated event (several might be possible), and attempt to fit whatever meaning it can extract from the utterance into that plan. Specifically, it evaluates the following communicative-act subtrees in order, depth-first (each has many descendants), and asks, "Could this be the intended communicative act?": REQUEST-ACTION, REQUEST-INFO, PROPOSE-PLAN, REPORT-PLAN-STATUS, REPORT-SYMPTOM, REPORT-HYPOTHESIS, REPORT-DIAGNOSIS, DESCRIBE-PLAN, DESCRIBE-CONCEPT, DESCRIBE-SCRIPT, DESCRIBE-INSTANCE.

Although this kind of decision-making can be challenging, our current model covers a non-trivial subset of cases. For example, an agent can infer that *A flathead screwdriver would be good* is a REQUEST-GIVE-OBJECT (a descendant of REQUEST-ACTION) because the detection heuristics for REQUEST-GIVE-OBJECT include (in plain English, not the ontological metalanguage): (a) the input expresses that some physical object would be useful as an INSTRUMENT for the given plan; (b) that object is available; and (c) the hearer is able to give it to the speaker. Similarly, the agent can infer that "The engine is too hot" is a REPORT-SYMPTOM-MECHANICAL (a descendant of REPORT-SYMPTOM) because the detection heuristics for REPORT-SYMPTOM-MECHANICAL are that the input expresses that the value of a scalar attribute that is relevant to the current plan is too high or too low.

A recap of section 2.1: We have just seen how the agent needs to reason deeply about discourse relations and communicative acts in order to prepare to assess whether its understanding of an input is actionable. This reasoning occurs at the tail end of input interpretation and the results are folded into one or multiple candidate meaning representations that the agent generates from its perceptual inputs.

2.2 Comparing Meaning Representations and Flagging Problems

The agent now reasons about its candidate meaning representations (there might be more than one) and records any specific analysis problems as flags appended to them. There are two kinds of flags: within-candidate flags and cross-candidate flags. Within-candidate flags involve a single meaning representation: for example, if the input includes an unknown word, every meaning representation will include a flag for this.[6] Cross-candidate flags result from the agent's comparison of different available analyses. For example, if the agent is not sure if "it" refers to the hammer or the screwdriver, then one candidate meaning representation will resolve it to the hammer and another will resolve it to the screwdriver. If the agent doesn't have a reason to prefer one analysis over the other, then each candidate will be appended with the flag "binary-ref-ambig[HAMMER, SCREWDRIVER]". By contrast, if the agent thinks that the hammer is the more likely referent, then the candidates will

[6] Agents do hypothesize the meanings of unknown words, but their analyses are typically rather vague. See LongGame (ch. 7) for details of autonomous learning of lexicon and ontology.

be flagged "binary-ref-ambig[prefer-HAMMER, disprefer-SCREWDRIVER]". Problem flags guide the agent's reasoning about actionability.

If there are no problem flags, this means that the agent has fully understood both the meaning and the intention (COMMUNICATIVE-ACT) of the input, so it can proceed to planning. It must be emphasized that confidently understanding the input does not necessarily mean that the agent can respond successfully—that is a matter of planning and execution, not perception interpretation. So, an agent might understand full well that "Go grab the vacuum" is intended as a request even if it cannot, for whatever reason, actually do it.

2.3 If the Application Doesn't Require Full Understanding…

Whether an agent is permitted to act with incomplete understanding of the input is a setting that depends upon the nature of application and/or the preferences of different users/collaborators. Whereas critical applications will likely allow no room for error, in non-critical ones, users will likely prefer for agents to take their best guesses rather than doublechecking too often.

If the agent is not permitted any errors, then any flags in the meaning representation will necessarily require it to seek clarification (cf. section 2.4). By contrast, if the agent *is* permitted to act with less than complete understanding, then its next step is potentially much more complicated: trying to determine whether it can reliably and usefully respond to some portion of the input.

As with other things we have discussed, this reasoning could be endlessly complex, considering that an input can include any number of fragments or sentences, they can present any number and combination of analysis problems, the application and/or the humans in the loop can be more and less forgiving of mistakes (even if they are permitted in principle), and so on. However, our current model serves as a viable starting point. According to it, the agent can consider its understanding actionable, even if incomplete or imperfect, if:

1. Some portion of the input includes a communicative act that requires an action as a response. The clearest examples are commands (requests for action) and questions (requests for information).[7] Contrast these with more passive communicative acts, like conveying information that the agent is just supposed to remember.
2. The theme of this communicative act—i.e., what is being requested or asked about—is fully understood by the agent: i.e., the agent understands what action needs to be carried out or what question needs to be answered.
3. The agent or somebody else in the team can, at least in principle, carry out the action.
4. The input does not include any warnings about or preconditions for the action that the agent doesn't fully understand.

Contrastive examples will show how this reasoning plays out.

[7] We constrain our examples to these two most common communicative acts in order to avoid delving into details about other communicative acts that are in bounds, such as proposing a plan, seeking a plan, and so on.

Example 1: A physician-in-training says to a virtual patient, *I need to know if you exercise daily because of the correlation between increased exercise and improved health outcomes*.

- The agent confidently interprets *I need to know if you exercise daily* as the communicative act REQUEST-INFO-YN-EPISODIC.
- It understands what it means to exercise daily (the theme of the request).
- It can look up the answer in its episodic memory and respond to the question.
- Even though the agent doesn't fully understand the reason for the question ("because of the correlation between increased exercise and improved health outcomes") this does not preclude it from responding.
- So, the agent will consider the input actionable and will proceed to planning its verbal action in response.

Example 2: A person says to a robot, *Open the engine cover. Use the screwdriver poking out of the tool box.*

- The agent confidently interprets *Open the engine cover* as the communicative-act REQUEST-ACTION.
- It understands what opening the cover is.
- It knows how to carry out this kind of action.
- However, it doesn't understand what *poking out of* means, which is information that relates to the INSTRUMENT of the action and, therefore, might be important.[8]
- So, the agent will not consider this input actionable and will necessarily ask a clarification question (rather than waiting and seeing) since it was explicitly asked to do something.

To generalize, we have designed LEIAs to take into account prerequisites and counterindications for actions, so they will not act before checking them. If the agent achieves an actionable but incomplete interpretation, it acts and also keeps a trace of what it did not understand, which it might choose to pursue later on. By contrast, if the agent does not achieve an actionable interpretation, it proceeds to deciding whether or not to seek clarification.

2.4 Seek Clarification or Not

When an agent can't achieve an actionable interpretation, its two main options are to wait and see or to ask a clarification question.[9] In the current version of the system, we are having LEIAs always

[8] The agent might have specific ontological knowledge about what tool is used to open an engine cover or it can apply more general knowledge that screwdrivers are tools, and tools are instruments of actions.

[9] Another option would be to consult an outside resource for help, such as a language model. However, large language models are too resource heavy for robots, small language models will be of questionable utility, preparing either to help with situational reasoning would be an entire research project in itself, and tinkering with language models is scientifically uninteresting, so our priorities lie elsewhere.

ask clarification questions, in keeping with the objectives of near-term applications. However, if LEIAs end up working in large teams of humans, they will need to be able to assess when to hang back since the humans might be talking to each other, not to them. The decision about when to seek clarification and when to wait and see will need to be based on evaluating parameters like the following:

- the number and kind (human vs. agent) of collaborators in the context: the more there are, the higher the chance that a difficult communication isn't aimed at the LEIA
- the preferences of the humans in the loop: do they prefer to field more clarification questions or will they allow the LEIA to make some mistakes?
- the urgency of the task: *wait and see* might not be an option in a time-sensitive application
- exactly what is and isn't understood: any time the agent directly receives a command or is asked a question, *wait and see* isn't an option—it has to act.

Since we have been focusing on having agents pursue clarification, we have developed a detailed model of what they ask. This depends on the particular problem flags that were issued during input interpretation, which reflect not only which difficulties were encountered, but also what the agent thinks the answer might be, if it has a reason to prefer one option over another. Examples of the ordered cases (which number in the dozens) are as follows:

- There is just one flag, it involves referential ambiguity, and the agent has a best guess. The agent doublechecks its best guess: "Bring Mandy the hammer." "You mean Mandy Smith, right?"
- There is just one flag, it involves referential ambiguity, and the agent doesn't have a best guess. It asks for clarification by stating the options it recognizes: "Do you mean Mandy Smith or Mandy Adams?"
- There is just one flag, it involves lexical ambiguity, and the agent has a best guess It doublechecks its best guess: "Scrub the computer before you shut it down." "Scrub means clear its data?"
- There is just one flag, it involves an unknown word, and the agent doesn't have a semantically specific guess as to its meaning. It asks for a definition: "Bring Richard the screwdriver I left in my locker." "What's a locker?"

The full inventory of eventualities covered by the model includes cases of multiple flags in various combinations. In some cases so many flags are present that the agent asks for a paraphrase instead of planning a long series of clarification questions. The objective of the clarification model is for the agent to get the information it needs in the most efficient way possible, so as not to annoy its human collaborators or impede the team's work overall.

3. Demonstrating the Actionability Assessment Model

The conference presentation will include a live demonstration of the actionability model in the DEKADE development and demonstration environment. DEKADE allows users to develop, debug,

and demonstrate LEIA cognition outside of a full simulation environment. In what follows we present a trace of this demo, interspersed with explanatory text.

To trace agent cognition in DEKADE, the user preloads situational parameters that the agent would have access to in a simulated or embodied system. Below is an excerpt (for reasons of space) from the description of the situation setup, exactly as presented in DEKADE, with somewhat simplified formatting for legibility. Underlined entities are ontological concept instances that can be clicked on to show their full descriptions; checkmarks indicate completed subtasks; bull's eyes indicate tasks in progress; and empty circles indicate future tasks on agenda.

> In this situation, the agent #LEIA.1 is on #TEAM.1 in a critical mission where small mistakes in interpretation are not allowed. The agent is in #ROOM.1, along with the speaker Samantha #HUMAN.1, Mary #HUMAN.2, #SCREWDRIVER.1, and #HAMMER.1.
>
> The agent's team is involved in #COLLABORATIVE-ACTIVITY.1 to perform maintenance on #ENGINE.1; the next task is to #REMOVE-PART.1 the stopper.

The plan on the agent's agenda (shown below) is a COLLABORATIVE-ACTIVITY whose THEME is MAINTENANCE. The plan has three subtasks: RUN-DIAGNOSTICS, REMOVE-PART (the stopper for the oil tank) and REPLACE-FLUID (the oil). The diagnostics have already been run and the next step is to remove the stopper.

```
#COLLABORATIVE-ACTIVITY.1
   #TEAM.1 is doing #MAINTENANCE.1
   √  #RUN-DIAGNOSTICS.1
   ⊙ #REMOVE-PART.1
   ○ #REPLACE-FLUID.1
REMOVE-PART.1
   STATUS          Status.NEXT
   AGENT           #HUMAN.2
   THEME           @STOPPER
   LOCATION        #ENGINE.1
   INSTRUMENT      #SCREWDRIVER.1
```

At this point, the developer types in, "Give that to Mary," providing no linguistic or extralinguistic clues to make clear the referent for "that". As indicated earlier, both a hammer and a screwdriver are in the room and the agent is aware of them (which can be confirmed by looking at the vision meaning representation, VMR, in its situation model, which is not shown here but will be demonstrated live at the conference). The trace of the agent's thinking at this moment is displayed:

```
I don't understand this well enough, I need clarification.
   Interpretation Flags
   "that" could refer to #HAMMER.1 but it probably refers to #SCREWDRIVER.1
```

The "Interpretation Flag" metadata shows why the agent reached this conclusion: it recognized the referential ambiguity of "that" but understood that the screwdriver was the more likely referent because a screwdriver is a necessary instrument for the current plan. The agent knows how to assess which candidate referent is most salient thanks to the procedural semantic routine that is called from the lexical sense of *that-n1*. The knowledge needed to support this reasoning was already displayed above: "REMOVE-PART.1 (INSTRUMENT #SCREWDRIVER.1)".

The agent selects the most appropriate clarification plan (shown in boldface below) from among the many options (shown in lighter shading) because this plan aligns with its state of knowledge: it has a best guess and seeks to doublecheck in the most efficient way possible.

Clarification Plan
- @DOUBLECHECK-COERCION
- @DOUBLECHECK-COMM-ACT
- @DOUBLECHECK-GLOBAL-INTERPRETATION
- @DOUBLECHECK-LEX-DISAMBIG
- @DOUBLECHECK-LEX-LEARNING
- **@DOUBLECHECK-REF-DISAMBIG**
- @REQUEST-BINARY-LEX-DISAMBIG
- @REQUEST-BINARY-REF-DISAMBIG
- @REQUEST-LEX-DISAMBIG
- @REQUEST-REF-DISAMBIG
- @SEEK-MORE-PRECISE-MEANING

The agent adds an instance of this plan to its agenda, which is now as follows:

AGENDA
- GOAL: MMR.1/AGENT-FUNCTIONING-FLOW.1 wants PLAN-STATUS to be COMPLETE
  - @AGENT-FUNCTIONING-FLOW
    - √ @PERCEPTION-RECOGNITION
    - ⊙ @PERCEPTION-INTERPRETATION
      - √ [BASIC]
      - ⊙ @SITUATIONAL-SEMANTICS-INTERPRETATION
        - ⊙ [CLARIFICATIONS]
          - ⊙ **@DOUBLECHECK-REF-DISAMBIG**
    - ○ @REASONING-ABOUT-ACTION
    - ○ @QUEUE-ACTION

The agent carries out this plan by asking:

The screwdriver, right?

Variations on this theme show different agent reasoning resulting from different situational parameters and ontological knowledge. For example, if the application has some tolerance for errors,

then in the scenario above the agent will simply give the screwdriver to Mary. By contrast, if the agent is given the same order ("Give that to Mary") but it doesn't know enough about the task at hand to understand which tool is needed, then it needs to ask the question "The hammer or the screwdriver?"

These examples are simple, perhaps surprisingly so. After all, we could have presented a very complex example and supplied the agent with all of the customized knowledge and reasoning it needed to make a demo work. But creating demos is not our objective. Our objective is to prepare agents to operate usefully in a wide variety of contexts using models, knowledge bases, implementations, and learning strategies that are as generic as possible. We believe that this is the most promising road to overcoming the scalability problem of content-centric systems (see McShane et al., 2025, for further discussion).

4. Can a LEIA Robot Actually Carry out the Action in Question?

The second use of the term *actionability* relates to whether the agent can successfully carry out the needed action in the given context. The modeling of this capability is ongoing, but the main principles can already be sketched out.

As we already explained, the perception interpretation actionability model leverages the agent's knowledge of its own capabilities without including all of planning and execution in the process of interpreting inputs. We gave the example of an agent that knows it can fetch things but knows it can't drive a vehicle. The way an agent knows whether it has a skill *in principle* is based on whether the associated event description in its ontology includes a filler for the property CALL-EFFECTOR.

CALL-EFFECTOR is attached to whatever grain-size of concept reflects the robot's actual implementation (the inventory of implemented effectors or policies and a control architecture). For example, if the whole process of fetching an object is implemented using a single function call, then that call will be attached to the concept FETCH-OBJECT, even if that concept is further described using subevents that support the agent's reasoning about what it means to fetch an object. This is shown below using a small excerpt from the ontological description of FETCH-OBJECT.

FETCH-OBJECT
  AGENT LEIA-#1
  THEME PHYSICAL-OBJECT-#1
  CALL-EFFECTOR *fun-fetch-object*
  HAS-EVENT-AS-PART LOCATE-OBJECT-#1, MOVE-TO-OBJECT-#1, etc.

By contrast, if the process of fetching an object is (as would be expected) divided into subfunctions on the robotic side, then the head of the script is not supplied with the CALL-EFFECTOR property but the subfunctions are.

FETCH-OBJECT
  AGENT LEIA-#1
  THEME PHYSICAL-OBJECT#1
  HAS-EVENT-AS-PART LOCATE-OBJECT-#1, MOVE-TO-OBJECT-#1, etc.
LOCATE-OBJECT-#1

AGENT	LEIA-#1
THEME	PHYSICAL-OBJECT#1
CALL-EFFECTOR	*fun-locate-obj*
MOVE-TO-OBJECT-#1	
AGENT	LEIA-#1
THEME	PHYSICAL-OBJECT#1
CALL-EFFECTOR	*fun-move-to-obj*

Functions listed as fillers of the CALL-EFFECTOR property address the conditions under which the agent can execute the skill and what the agent's physical limitations are. This allows LEIAs to determine whether an action is likely to be feasible before attempting execution. This static knowledge gives LEIAs useful information about their own capabilities when they are trying to make sense of what a collaborator said (i.e., it *is* part of the actionability model detailed in this paper) without subsuming all of planning into the process of input interpretation.

After input interpretation, when the LEIA thinks it knows what is expected of it, it carries out additional analysis to determine whether the given action is feasible in the given context. This can include simulation or the use of formal verification models, such as Jacobian singularity checking or Lyapunov stability analysis.

When situation-specific verification indicates that an action cannot be executed—for example, due to reachability constraints, collision risks, or other physical limitations—this failure information is fed back to the cognitive layer of the LEIA. The state representation in the situation model is updated with why the action would fail, which allows the LEIA to select its next move from general strategies available in its COLLABORATIVE-ACTION script (e.g., report that you can't do it, report why, suggest a different plan) without having to attempt an action that is sure to fail.

The final way a LEIA robot can assess whether it can carry out an action is to attempt it and see what happens. Failure detection relies on multiple signals: the environment state does not change as predicted, proprioceptive feedback indicates problems, or post-execution perception confirms the goal was not achieved. This all requires robust environment state tracking to compare expected versus observed outcomes. When execution fails, this becomes new perceptual input: the system learns that certain actions are not actionable under particular conditions, informing future reasoning about which actions to attempt.

There are more details on the robotic side that take us still farther away from our main thesis but might be of interest to some readers. For example, it is possible to have calls to physical actions recorded at multiple levels of an ontological subtree. We will consider the case of picking up objects. Picking up an object from the floor and picking up an object from a shelf involve different robotic policies. However they are both descendants of a more generic notion of picking something up from wherever it might be located. So, the *pickup()*, function can have one argument or two, as shown below.

PICKUP	
AGENT	LEIA-#1
THEME	PHYSICAL-OBJECT-1
CALL-EFFECTOR	pickup(callID)

```
PICKUP-FROM-FLOOR                    ; AGENT & THEME as above
   SOURCE              FLOOR
   CALL-EFFECTOR       pickup(callID,FLOOR-#1)
PICKUP-FROM-SURFACE                  ; AGENT & THEME as above
   SOURCE              FLAT-SURFACE
   CALL-EFFECTOR       pickup(callID,FLAT-SURFACE-#1)
```

The above approach to knowledge engineering and system implementation supports a robot's picking up objects from unspecified places. When the input parameters do not specify where an object is located, control of the process is determined by the tactical layer, not the cognitive layer. For example, if a LEIA robot is told to pick up the thermostat but is not told *from where*, the low-level planner that guides the robot's movements executes pickup from the floor first. The attention service on the tactical layer monitors the result, expecting a thermostat in hand. If there is no thermostat in hand after the first attempt, the system immediately executes pickup from the shelf. This sequencing happens entirely at the tactical level, with the same action command reparameterized based on execution outcomes, requiring no strategic-level intervention.

5. Comparisons with Others

We know of no other cognitive models of actionability that would serve as direct points of comparison with the model presented here. As regards broader comparisons with relevant subfields of AI, they are detailed in LingAI and LongGame, which are available open access for interested readers. The most relevant language-related work dates back quite a while. For example, in his work on dialog acts, Traum (2000, p. 7) said, "When engaging in a study related to dialogue pragmatics, a researcher is confronted with a bewildering range of theories and taxonomies of dialogue acts to choose from." As is common in theoretical and descriptive linguistics, dialog acts have been neatly shaved off from two intimately connected phenomena: the meaning of the propositions scoped over by those dialog acts, and any actions outside of language that are relevant to the situation, such as responding to an utterance by shrugging. So, dialog act models address the fact that when someone asks a question, the interlocutor typically answers it, but they say nothing about what is actually asked. As Traum (1999, p. 1) writes, "In studying speech acts, the focus is on pragmatics rather than semantics – that is, how language is used by agents, not what the messages themselves mean…" As for dialog modeling, between around 1980 and the early aughts, it was studied in earnest, primarily as an aspect of planning. For example, Allen & Perrault (1980) put forth a goal- and plan-based approach to dialog processing, influenced by classical AI approaches to planning. Later work in dialog processing (e.g., Lemon & Gruenstein, 2004) shifted to relying predominantly on dialog cues – still isolated from semantic content.

6. Discussion

One of the challenges in developing cognitive-robotic systems is ordering priorities, which makes it reasonable to ask whether dealing with the actionability of input interpretation—i.e., overtly anticipating that the agent might not fully understand what it perceives—should make it to the near-

term agenda. We think yes, but with our usual caveat: the model and its implementation should cover the kinds of eventualities that will actually be encountered by agent systems, not everything that an overenthusiastic descriptive linguist or psychologist could imagine. The goal is to keep people from getting frustrated with, and therefore rejecting, agent systems because they make what appear to be outrageously stupid mistakes. The problem is that people don't realize how much reasoning content-based systems have to do when interpreting the world, so they don't understand the challenges facing computer systems trying to match human behavior. So we, as developers, need to anticipate fail points and try to engineer our way around them, at all times remembering that our agents must be not only be capable but also transparent, explanatory and trustworthy.

Although the model presented here accommodates multimodal stimuli, the paper has talked primarily about language for two reasons. First, some of the phenomena, like indirect speech acts, are specific to language. Second, whereas a LEIA's language processing remains consistent across different simulated and embodied environments, the interpretation of visual, haptic, and sensor inputs will play out differently—in ways we are currently exploring in collaboration with roboticists.

In today's climate of excitement over language models, an obvious question is whether language models couldn't somehow help with the cluster of problems described here. The short answer is yes. For example, a language model can sometimes suggest the elided semantic relationship between pairs of propositions. When we asked ChatGPT's (on Aug. 17, 2025), "Tell me in five words or less what the semantic relationship is between 'I'm changing the tire' and 'I ran over a nail'". Its response was "Cause-and-effect relationship," which is correct and useful. Of course, the deficiencies of language models are too well reported to bear repeating, so it would be unwise to put too much faith in them. In addition, *large* language models are of little use in robots, and it is unclear whether *small* language models would have sufficient coverage to be useful. In short, we are exploring various ways of integrating language models as tools into a LEIA's lifelong learning and runtime processing environment. We will report the results of this work at a later date.

Acknowledgements

This research was supported in part by Grants N00014-23-1-2006 and N00014-24-1-2679 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

Allen, J. F., & Perrault, C. R. (1980). A plan-based analysis of indirect speech acts. *Computational Linguistics* 6(3-4): 167–182.

Bailer-Jones, D. M. (2009). *Scientific models in philosophy of science*. University of Pitts- burgh Press.

Carlson, L., Marcu, D., & Okurowski, M. E. (2003). Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In J. van Kuppevelt & R. W. Smith (Eds.), *Current and new directions in discourse and dialogue* (pp. 85–112). Kluwer.

Das, D., & Taboada, M. (2018). RST signalling corpus: A corpus of signals of coherence relations. *Language Resources and Evaluation*, *52*(1), 149–184. https://doi.org/10 .1007/s10579-017-9383-x

Lemon, O., & Gruenstein, A. (2004). Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction*, 11(3): 241–267.

Marchal, M., Scholman, M. C. J., & Demberg, V. (2020). The effect of domain knowledge on discourse relation inferences: Relation marking and interpretation strategies. *Dialogue & Discourse*, *13*(2), 49–78. https://doi.org/10.5210/dad.2022.202

McShane, M., & Nirenburg, S. (2021). *Linguistics for the Age of AI*. The MIT Press. [LingAI]

McShane, M., Nirenburg, S., & English, J. (2024). *Agents in the Long Game of AI: Computational cognitive modeling for trustworthy, hybrid AI*. MIT Press. [LongGame]

Nirenburg, S., & Raskin, V. (2004). *Ontological Semantics*. MIT Press.

Oruganti, S., Nirenburg, S., McShane, M., English, J., Roberts, M., Arndt, C. 2024a. HARMONIC: A framework for explanatory cognitive robots. *Proceedings of ICRA@40*. Rotterdam, The Netherlands, September.

Oruganti, S., Nirenburg, S., McShane, M., English, J., Roberts, M. K., & Arndt, C. (2024b). HARMONIC: Cognitive and Control Collaboration in Human-Robotic Teams. *arXiv preprint arXiv:2409.18047.*

Traum, D. R. (1999). Speech acts for dialogue agents. In M. Wooldridge & A. Rao (Eds.), *Foundations and Theories of Rational Agents* (pp. 169–201). Kluwer.

Traum, D. R. (2000). 20 Questions for Dialogue Act Taxonomies. *Journal of Semantics*, 17(1):7–30.