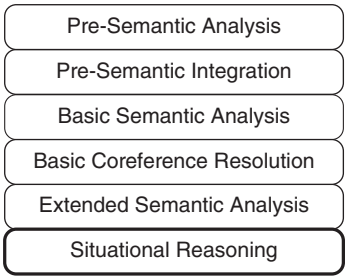
Linguistics for the Age of AI

The title-level DOI for this work is:

doi:10.7551/mitpress/13618.001.0001

7

Situational Reasoning



Up until this point, all of the LEIA's language analysis methods have been generic: they have leveraged the agent's lexicon and ontology and have not relied on specialized domain knowledge or the agent's understanding of what role it is playing in a particular real-world activity.

Considering that LEIAs are modeled after people, one might ask, "Don't people—and, therefore, shouldn't LEIAs—always know what context they are in? And, therefore, shouldn't this knowledge always be the starting point for language analysis rather than a late-stage supplement?" Although there is something to be said for this observation, the fact is that people actually can't always predict what their interlocutor's next utterance will be about. For example, while making dinner, you can be talking not only about food preparation but also about the need to buy a new lawn mower and a recent phone message that you forgot to mention. In fact, people shift topics so fast and frequently that questions like "Wait, what are we talking about?" are not unusual. The ubiquity of topic switching provides a theoretically motivated reason to begin the process of NLU with more generic methods—since the given utterance might be introducing a new topic—and then invoke context-specific ones if needed.

In addition to theoretical motivations, there are practical motivations for starting with generic methods of NLU and progressing to domain-specific ones as needed.

1. Ultimately, agents need to be able to operate at a human level in all contexts. Therefore, NLU capabilities should be developed in a maximally domain-neutral manner.
2. Many types of domain-specific reasoning are more complicated and computationally expensive than the generic NLU described so far, since they often involve capabilities like inferencing and mindreading.[1] Therefore, they should be used only if needed.
3. It can be difficult to automatically detect the current topic of conversation since objects and events from many domains can be mentioned in a single breath. Moreover, it is not the case that every mention of something shifts the topic of conversation to that domain. For example, saying, "Look, the neighbor kid just hopped over the fence for his baseball," does not mean that everyone should now be poised to think about strikes and home runs.

4. Although developers can, of course, assert that an agent will operate in a given domain throughout an application (and LEIAs can be manipulated to function that way as well), associated successes must be interpreted in the light of this substantial simplification. After all, committing to a particular domain largely circumvents the need for lexical disambiguation, which is one of the most difficult challenges of NLU.

Since Situational Reasoning is grounded in a particular domain and application, we might be tempted to delve directly into system descriptions. That will come soon enough, as applications are the topic of the next chapter. In this chapter we will (a) say a few more words about the OntoAgent cognitive architecture, all of whose components are available for Situational Reasoning, and (b) introduce our general model of integrating situational knowledge and reasoning into natural language understanding.

7.1 The OntoAgent Cognitive Architecture

As we have been demonstrating throughout, natural language understanding is a reasoning-heavy enterprise, and there is no clear line between reasoning *for* language understanding and reasoning *beyond* language understanding. We couldn't agree more with Ray Jackendoff's (2007) opinion that we cannot, as linguists, draw a tight circle around what some call *linguistic meaning* and expect all other aspects of meaning to be taken care of by someone else. He writes:

> If linguists don't do it [deal with the complexity of world knowledge and how language connects with perception], it isn't as if psychologists are going to step in and take care of it for us. At the moment, only linguists (and to some extent philosophers) have any grasp of the complexity of meaning; in all the other disciplines, meaning is reduced at best to a toy system, often lacking structure altogether. Naturally, it's daunting to take on a problem of this size. But the potential rewards are great: if anything in linguistics is the holy grail, the key to human nature, this is it. (p. 257)

It is only for practical purposes that this book concentrates primarily on the linguistic angles of NLU; doing otherwise would have required multiple volumes. However, as we introduced in chapter 1, full NLU is possible only by LEIAs that are modeled using a full cognitive architecture. The cognitive architecture that accommodates LEIAs is called OntoAgent. Its top-level structure is shown in figure 7.1.

This architecture is comprised of the following components:

- Two input-oriented components: perception and interpretation;
- The internal component covering attention and reasoning;
- Two output-oriented components: action specification and rendering; and
- A supporting service component: memory and knowledge management.

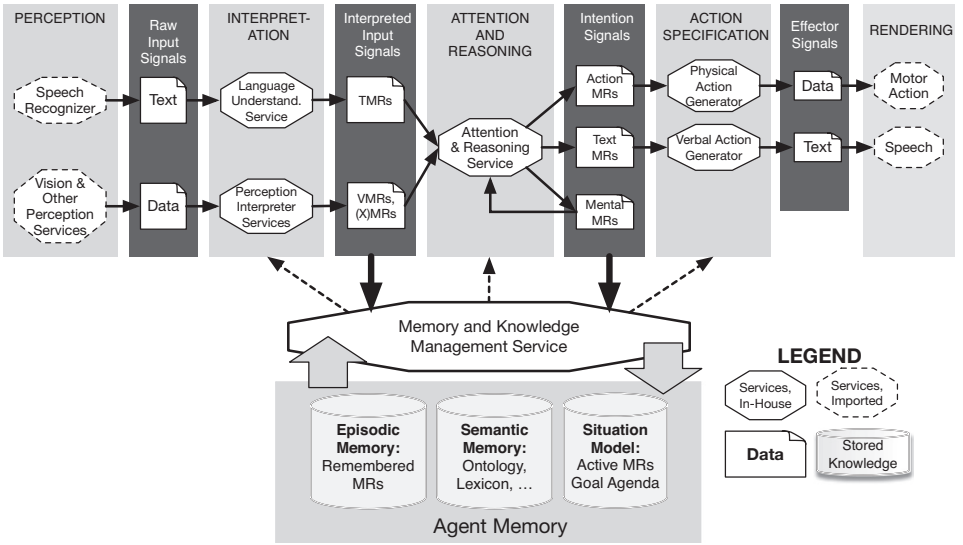


Figure 7.1
A more detailed view of the OntoAgent architecture than the one presented in figure 1.1.

The NLU capabilities described in this book are encapsulated in the Language Understanding Service. Just as it interprets textual inputs (which are, in some cases, transcribed by a speech recognizer), the Perception Interpreter Services must interpret other types of perceptual inputs—vision, nonlinguistic sound, haptic sensory inputs, and even interoception, which is the agent's perception of its own bodily signals. Whereas the NLU processes described so far relied on the agent's Semantic Memory (ontology and lexicon), the Situational Reasoning it now brings to bear can rely on its Situation Model and Episodic Memory as well.

This chapter continues to concentrate on our objective of describing the program of R&D that we call Linguistics for the Age of AI by discussing methods for treating the following phenomena using all the resources available to LEIAs operating in a situated context: (a) fractured syntax, (b) residual lexical ambiguity, (c) residual speech-act ambiguity, (d) underspecified known expressions, (e) underspecified unknown word analysis, (f) situational reference, and (g) residual hidden meanings.

7.2 Fractured Syntax

As described in section 3.2 and illustrated by figure 3.4, if syntactic mapping (i.e., the process of aligning elements of the parse with word senses in the lexicon) does not work out perfectly, the agent can choose to either (a) establish the best syntactic mapping it can and

continue through the canonical middle stages of NLU (Basic Semantic Analysis, Basic Coreference Resolution, and Extended Semantic Analysis) or (b) circumvent those stages and proceed directly to this stage, where it will attempt to compute a semantic interpretation with minimal reliance on syntax. We call the methods it uses for the latter *fishing* and *fleshing out*. Fishing is used for wordy inputs: it involves extracting constituents that semantically fit together while potentially leaving others unanalyzed. Fleshing out, by contrast, is used for fragmentary, telegraphic utterances: it involves filling in the blanks given the minimal overt constituents. The reason why fishing and fleshing out are postponed until Situational Reasoning is that the agent needs some knowledge of what is going on to guide the analysis. In the absence of context, even people cannot make sense of highly fractured utterances.

The functions for fishing and fleshing out are applied in sequence. The fishing algorithm performs the following operations:

1. It strips syntactic irregularities—mostly, production errors such as repetitions and self-corrections—using more sophisticated detection algorithms than were invoked during Pre-Semantic Integration. For example, if two entities of the same syntactic category are joined by one of our listed self-correction indicators (e.g., *no*; *no, wait*; *no, better*), then strip the first entity along with the self-correction indicator: for example, *Give me the wrench, no, better, the screwdriver* → *Give me the screwdriver.*
2. It identifies NP chunks in the parse, since even for syntactically nonnormative inputs, NP chunking tends to be reliable enough to be useful.
3. It generates the most probable semantic analyses of the individual NP chunks on the basis of their constituents (e.g., the best semantic correlation between an adjective and its head noun) as well as domain- and context-oriented preferences. For example, if the agent is building a chair, then the preferred interpretation of *chair* will be CHAIR-FURNITURE not CHAIRPERSON. As we said earlier, the agent has to have some understanding of the domain/context in which it is operating in order for the fishing process to have a fair chance of working.
4. It uses the preferred NP analyses generated at step 3 as case role fillers for the events represented by verbs in the input. The disambiguation of the verbs is guided by both the meanings of the surrounding NPs and the most likely meaning of the verb in the given domain/context. Continuing with our chair-building example, the verb *give* has a preferred meaning of TRANSFER-POSSESSION, not DONATE or any of the other candidate analyses of that word.
5. It attempts to account for residual elements of input—such as modals, aspectuals, negation, and adverbials—by attaching the candidate analyses of residual elements to the just-generated TMR chunks, using, essentially, an *ordered bag of concepts* approach. For example, if the modal verb *should* occurs directly to the left of a verb whose

preferred analysis (from step 4) is TRANSFER-POSSESSION, then the meaning of *should*—that is, obligative MODALITY with a value of 1—will scope over that instance of TRANSFER-POSSESSION.

If fishing does not yield an actionable result, the agent can attempt fleshing out. In this process, the agent asks itself, "Could this (partial) meaning representation I just generated actually be a question I can answer? Or a command I can carry out? Or a piece of information I need to remember?" An example will best illustrate the idea.

Assume that a person who is collaboratively building a chair with a robot says, "Now the seat onto the base," which is an elliptical utterance that lacks a verb (semantically, an EVENT). Among the candidate interpretations of *the seat* and *the base* are CHAIR-SEAT and FURNITURE-BASE, which are concepts that figure prominently in the BUILD-CHAIR script. The use of these highly relevant concepts (a) triggers the agent's decision to go ahead and pursue fleshing out and (b) allows the agent to commit to these interpretations of *seat* and *base*.

Now the agent must analyze the other two contentful words in the input: *now* and *onto*. In the context of instruction giving, sentence-initial *now* can indicate the speech act REQUEST-ACTION, so the agent can assume that it is being asked to do something. *Onto*, for its part, means DESTINATION when applied to physical objects. So, on the basis of the co-occurrence of the above possibilities, as well as the ordering of words in the input, the agent can hypothesize that (a) it is being called to do some action, (b) the action involves as its THEME a specific instance of CHAIR-SEAT, and (c) the DESTINATION of that action is a particular instance of FURNITURE-BASE. This results in the initial TMR:

REQUEST-ACTION-1
- AGENT — HUMAN-1 ("speaker")
- THEME — EVENT-1
- BENEFICIARY — ROBOT-1

EVENT-1
- THEME — CHAIR-SEAT-1
- DESTINATION — FURNITURE-BASE-1
- COREF — *seek-specification*

The procedural semantic routine *seek-specification* is included in the EVENT-1 frame because the agent hypothesized that some event was elliptically referred to, and now it has to try to figure out which one. It hits pay dirt when it finds, in its BUILD-CHAIR script, the subevent

ATTACH
- THEME — CHAIR-SEAT
- DESTINATION — FURNITURE-BASE

It can then replace the unspecified EVENT from its initial TMR with ATTACH, leading to the following fleshed-out and actionable TMR.

REQUEST-ACTION-1
- AGENT HUMAN-1 ("speaker")
- THEME ATTACH-1
- BENEFICIARY ROBOT-1

ATTACH-1
- AGENT ROBOT-1
- THEME CHAIR-SEAT-1
- DESTINATION FURNITURE-BASE-1

The algorithms for fishing and fleshing out are actually much more complicated than what we just described. The reason we choose not to detail those algorithms here is that most of the reasoning is extralinguistic and depends centrally on what ontological knowledge is available for the given domain, what the agent's goals are, what its understanding of its interlocutor's goals are, what it is physically capable of doing, and so on. All of this is more appropriately presented within a comprehensive description of an agent application. If this sounds like a hint at our next book, it is.

7.3 Residual Lexical Ambiguity: Domain-Based Preferences

It often happens that Basic Semantic Analysis generates multiple candidate interpretations of an input. In some cases, Basic Coreference Resolution or Extended Semantic Analysis provides definitive evidence to select among them. But often, the agent reaches the stage of Situational Reasoning with multiple candidates still viable. This residual ambiguity can be treated in a straightforward manner by preferring analyses that use domain-relevant concepts. If the LEIA is building a chair, then the associated script will include many instances of CHAIR-FURNITURE and none of CHAIRPERSON. Accordingly, "I like chairs" (which allows for either interpretation on general principles) should be analyzed using CHAIR-FURNITURE. This approach might remind one of distributional semantics, with the crucial difference that, for our agents, meaning representations contain unambiguous concepts rather than ambiguous words.

7.4 Residual Speech Act Ambiguity

Recall that utterances that offer an indirect speech act reading always offer a direct speech act meaning as well. "I need a hammer" can mean *Give me one* or simply *I need one*—maybe I know that you aren't in a position to give me one or that we don't have one to begin with. During Basic Semantic Analysis the LEIA detected the availability of both interpretations for typical formulations of indirect speech acts, which are recorded as pairs of senses in the lexicon. It gave a scoring preference to the indirect reading, but the direct meaning remained available. Now it can use reasoning to make the final decision. If the indirect interpretation is something that the agent can actually respond to—if it is a question the

LEIA can answer or a request for action that the LEIA can fulfill—then that interpretation is selected. This would be the case, for example, if a chair-building LEIA were told, "I need a hammer" (a request: *Give me a hammer*) or "I wonder if we have any more nails" (which could be a question: *Do we have any more nails?* or a request: *Give me some nails*). By contrast, if the LEIA cannot fulfill the request ("I would love a sandwich right now") or does not know the answer to a question ("I wonder why they need so many chairs built today"), then it chooses the direct interpretation and does not attempt any action.

7.5 Underspecified Known Expressions

What does *good* mean? If a vague answer is enough—and it often is—then *good* indicates a positive evaluation of something. In fact, the independent statements, "Good," "Great," "Excellent," and others have lexical senses whose meaning is "The speaker is (highly) satisfied with the state of affairs." This is important information for a task-oriented LEIA because it implies that no reparative action is needed. By contrast, "What a mess" and "This looks awful" indicate that the speaker is unhappy and that the LEIA might consider doing something about it.

However, there are cases in which a vague expression actually carries a more specific meaning. If I ask someone who knows me well to recommend a *good restaurant*, then I expect him or her to take my location, budget, and preferences for food and décor into account. If someone recommends a *good student* for a graduate program, that student had better be sharp, well prepared, and diligent. And the features contributing to a *good résumé* will be very different for people researching résumé design principles than for bosses seeking new hires. Similarly, as we saw in section 6.3.3, underspecified comparisons—such as *My car is better than this one*—may or may not require that some particular property value(s) be understood.

For applications like recommendation systems, web search engines, and product ratings, notions of *good* and *bad* span populations (they do not focus on individuals) and tend to generalize across features (e.g., a restaurant might get an overall rating of 3 out of 5 even though the food is exceptional). By contrast, in agent applications, the features of individuals—including their preferences, character traits, and mental and physical states—can be key. For example, in the domain of clinical medicine, on which we have worked extensively (see chapter 8), the best treatment for a patient will depend on a range of factors that the LEIA knows about thanks to a combination of its model of clinical knowledge and the features it has learned about the individual during simulation runs (through, e.g., dialog and simulated events). Clearly, this is all highly specific to particular domains, situations, and applications.

7.6 Underspecified Unknown Word Analysis

Up to this point, unknown words have been treated as follows. During Pre-Semantic Integration they were provided with one or more candidate lexical senses that were syntactically

specific but semantically underspecified. Then, during Basic Semantic Analysis, the semantic interpretation was narrowed to the extent possible using the unidirectional application of selectional constraints recorded in the ontology. For example, assuming that *kumquat* is an unknown word, given the input, *Kerry ate a kumquat*, the LEIA will understand it to be a FOOD because of its ontological knowledge about possible THEMEs of INGEST. Now the question is, *Can knowledge of, and reasoning specific to, a particular domain narrow the interpretation still further?* In some cases, it can.

Let's assume that the agent is operating in a furniture-building domain and receives the utterance "Pass me the Phillips." Let's assume further that it does not have a lexical sense for *Phillips*, or even the full-form *Phillips-head screwdriver*. The agent can narrow down its interpretation to the set of objects that it is able to pass, under the assumption that its interlocutor is operating under sincerity conditions. If the agent understands which action the human is attempting to carry out—something that can be provided either by language ("I'm trying to screw in this screw") or through visual perception—it can narrow the interpretation still further.

7.7 Situational Reference

The objective of situational reference is to anchor all the referring expressions (RefExes) a LEIA encounters to the corresponding object and event instances in its situation model. At this stage of the process, reference resolution transcends the bounds of language and incorporates RefExes obtained as a result of the operation of perception modalities other than language (see figure 7.1). Resolving these extended coreferences is called *grounding* in agent systems. (For other work on grounding see, e.g., Pustejovsky et al., 2017.) Multichannel grounding is likely to require a nontrivial engineering effort because most agent systems will need to import at least some external perception services, whose results need to be interpreted (using the Perception Interpreter Services) and then translated into the same ontological metalanguage used for the agent's knowledge bases and reasoning functions. Stated plainly, developers of integrated agent systems cannot develop all functionalities in-house; they need to incorporate systems developed by experts specializing in all aspects of perception, action, and reasoning; and few R&D teams include all these types of specialists. After object and event instances have been interpreted and grounded, it is a different decision whether or not to store them to episodic memory.

At this stage of NLU, Situational Reasoning, three reference-related processes occur, all in service of the grounding just described: (a) the agent vets the correctness of previously identified sponsors for RefExes using the situational knowledge that is now available, (b) it identifies sponsors in the linguistic or real-world context for RefExes as yet lacking a sponsor, and (c) it anchors the meaning representations associated with all RefExes in the agent's memory. We consider these in turn.

7.7.1 Vetting Previously Identified Linguistic Sponsors for RefExes

Let us begin by recapping coreference processing to this point. Sponsors for many RefExes have been identified using methods that are largely lexico-syntactic. The only kind of semantic knowledge leveraged so far has involved CONCEPT-PROPERTY-FILLER triples recorded for the open-domain ontology—that is, not limited to particular domains for which the agent has been specially prepared. To repeat just two examples from Basic Coreference Resolution: (a) the property HAS-OBJECT-AS-PART provides heuristics for detecting bridging constructions (e.g., ROOM (HAS-OBJECT-AS-PART WINDOW)), and (b) the *default* fillers of case roles suggest preferences for pronoun resolution (e.g., the AGENT of SURGERY should best be a SURGEON).

Now, at this stage, the agent incorporates additional knowledge bases and reasoning to determine whether previously posted coreference decisions are correct. The process incorporates (a) the agent's knowledge/memories of contextually relevant object and event instances; (b) its ontological knowledge, recorded in scripts, of the typical events in the particular domain in which it is operating; and (c) its understanding of what, exactly, it and its human interlocutor are doing at the time of the utterance. The vetting process, as currently modeled, is organized as the following series of five checks.

Check 1. *Do stable properties unify?* We define *stable properties* as those whose values are not expected to change too often. For people, these include MARITAL-STATUS, HEIGHT, HAS-SPOUSE, HAS-PARENT, and so on. For physical objects, they include COLOR, MADE-OF, HAS-OBJECT-AS-PART, and so on. At this point in microtheory development, we are experimenting with *an* inventory of stable properties without assuming it to be the optimal one, and we are well aware of the changeability of practically any feature of any object or event given the right circumstances or the passing of a sufficient amount of time. What this check attempts to capture is the fact that a blue car is probably not coreferential with a red one, and a 6′3″ man is probably not coreferential with a 5′2″ one.

Formally, the agent must first identify which entities in its memory are worth comparing with the entity under analysis; then it must check the value of the relevant property to see if it aligns. The problem, of course, is that although a feature-value conflict can suggest a lack of coreference, lack of a conflict does not ensure coreference. Consider some examples:

(7.1) John doesn't like Rudolph because he's 6′2″ tall.

- CoreNLPCoref corefers *he* and *Rudolph*, which is probably what is intended, but there is no way for the LEIA to know that, since the engine's overall precision in resolving third-person coreference is not extremely high.
- The LEIA checks its memory for Rudolph's height. If his HEIGHT is 6′2″, then the LEIA confirms that the coreference link could be correct—but it need not be, since John could also be 6′2″. If Rudolph's height is known and is something other than

6′2″, then the agent rejects the coreference link. If it doesn't know Rudolph's height, or can't find any Rudolph in its memory, then this check abstains.

(7.2) Madeline would prefer not to barhop with Justine because she's married.

- CoreNLPCoref corefers *she* and *Madeline*, which may or may not be what is intended (this sentence equally allows for either interpretation of the coreference).
- The LEIA checks whether it knows Madeline's marital status. If Madeline is married, then it confirms that the coreference link could be correct, even though it is possible that both women are married. If Madeline is not married, then it rejects the coreference link. In all other cases, this check abstains.

To recap, feature checking cannot confidently assert that a coreference link is correct, but it can exclude some candidate coreference links when the entities' feature values do not unify.

Check 2. *Can known SOCIAL-ROLEs guide sponsor preferences?* Prior knowledge of people's social roles can help to confirm or overturn previously posited coreference links. For example, given the following inputs and sufficient background knowledge about the individuals in question, the LEIA should be able to confirm their social roles.

(7.3) [The HUMAN referred to by *he* should have the SOCIAL-ROLE PRESIDENT.]
President George H. W. Bush offered "a kinder, gentler" politics. He lasted one term. Clinton called himself "a New Democrat." He got impeached. (COCA)

(7.4) [The HUMAN referred to by *he* should have the SOCIAL-ROLE SURGEON.]
Last week he operated on an infant flown in from Abu Dhabi. (COCA)

Another example in which this check can prove useful is our example from section 5.2.3: *Mike talked at length with the surgeon before he started the operation*. Upstream analysis suggested that *he* should corefer with *the surgeon* on the basis of ontological expectations—and that is true. But we also pointed out that Mike could be an anesthesiologist or a general practitioner preparing to carry out a minor surgery in his office. In the latter case, there is bona fide ambiguity without knowing more about the people involved in the context.

In considering how best to use social roles to guide coreference assignments, there is one important decision to be made: Should the social roles in question be constrained to those mentioned in the given discourse, or should any known social role found in the agent's memory about this individual be invoked? There is no simple answer. People typically have more than one social role. For example, someone whose profession is TEACHER can have any number of other social roles that are more contextually relevant, such as PARENT, COACH, HOMEOWNER, and so on.

Check 3. *Do the case role fillers of known EVENTS guide sponsor preferences?* This check will fire only if event coreference was already established—for example, due to a

repetition structure. In that case, the agent will prefer that the case roles of coreferential events have parallel fillers (i.e., the same AGENTS, the same THEMES, and so on). Consider in this regard example (7.5).

(7.5) "Roy hit Dennis after Malcolm told off Lawrence." "Why did he hit him?"

- CoreNLPCoref corefers *Malcolm*, *he*, and *him*, which is incorrect.
- The LEIA, instead, establishes the coreference between the instances of *hit* and then lines up their case role fillers: Roy_1 hit_2 Dennis_3 / he_1 hit_2 him_3.

Check 4. *Can domain-specific ontological knowledge guide sponsor preferences?* Let us return to the chair-building domain. If *it* in the utterance *Hit it hard* could refer to either a NAIL or a CHAIR-BACK, and if the furniture-building scripts include many instances of hitting nails and none of hitting chair backs, then the preferred resolution of *it* will be NAIL. (Of course, one *can* need to hit a chair back, but both a human and an agent would be best advised to double-check the speaker's meaning before doing that.)

Yet another case in which domain-specific ontological knowledge can guide sponsor selection involves elided events that, until now, have remained underspecified. Consider the example *Help me*, which is recorded as a lexical sense that detects the elided event (help you *do what*?) but requires situational reasoning to resolve it. The LEIA needs to determine which event in the script its collaborator is pursuing and whether or not it (the LEIA) can assist with it. Of course, detecting its collaborator's current activity requires sensory inputs of a kind we have not yet discussed (see chapter 8), but the basic principle should be clear. Naturally, knowing which subevent of the script is currently being pursued narrows the search space and increases the agent's confidence in its interpretation.

Check 5. *Can some aspect of general knowledge about the world guide sponsor preferences?* This is the class of phenomena illustrated by the Winograd challenge problems (Levesque et al., 2012).[2] For example:

(7.6) a. The trophy doesn't fit into the brown suitcase because it is too large.
b. The trophy doesn't fit into the brown suitcase because it is too small.

(7.7) a. Joan made sure to thank Susan for all the help she had received.
b. Joan made sure to thank Susan for all the help she had given.

(7.8) a. Paul tried to call George on the phone, but he wasn't successful.
b. Paul tried to call George on the phone, but he wasn't available.

The knowledge necessary to support such reasoning can be recorded in the ontology in a straightforward manner: a precondition for A being INSIDE-OF B is that A is smaller than B; a typical sequence of events (a tiny script) is A HELPS B and then B THANKS A; when one tries to do something, one can either succeed or fail; in order for a

COMMUNICATION-EVENT (like calling) to be successful, the person contacted must be available. No doubt, a lot of such knowledge is needed to support human-level reasoning about all domains—something pursued, for example, in the Cyc ontology acquisition effort (Lenat, 1995). However, since for the foreseeable future LEIAs will have this depth of knowledge only for specialized domains, this kind of reasoning is assigned to the current module, all of whose functionalities require knowledge support beyond what is available for the open domain.

7.7.2 Identifying Sponsors for Remaining RefExes

Some of the RefExes that do not yet have a sponsor need to be directly grounded in the physical context.[3] This kind of reference resolution is, strictly speaking, outside the scope of this book. So here we will just briefly comment on how agents interpret nonlinguistic percepts in order to accomplish the physical grounding of objects and events.

As we explained with respect to figure 1.1 in chapter 1, no matter how a LEIA perceives a stimulus—via language, vision, haptics, or otherwise—it must interpret it and record that interpretation in the ontologically grounded metalanguage. The results are stored in knowledge structures that we call XMRs: meaning representations (MRs) of type X, with X being a variable. When the input is text, the XMR is realized as a TMR (a *text* meaning representation), whereas when the input is vision, the XMR is realized as a VMR (a *visual* meaning representation), and so on (see figure 7.1).

All XMRs have a set of generic properties as well as a set of properties specific to their source. The VMR below grounds the event of assembling along with all of the objects filling its case roles. Specifically, it expresses the situation in which the robot has seen its human collaborator assemble a chair leg using a bracket and a dowel. Although the formalism looks somewhat different from the pretty-printed TMRs we have been presenting throughout, it is actually entirely compatible.

```
@INPUTS.VMR.1 = {
    INSTANCE-OF             @ONT.VMR;
    REFERS-TO-GRAPH         "VMR#1";
    STATUS                  "UNDERSTOOD";
    SIGNAL                  "INPUT";
    TYPE                    "VISUAL";
    TIMESTAMP               1549721130;
    SOURCE                  @SELF.ROBOT.1;
    ROOT                    @VMR#1.ASSEMBLE.1;
};
@VMR#1.ASSEMBLE.1={
    INSTANCE-OF             @ONT.ASSEMBLE;
    AGENT                   @ENV.HUMAN.1;
    THEME                   @ENV.ARTIFACT-LEG.1;
```

```
    INSTRUMENT          @ENV.BRACKET.1;
    INSTRUMENT          @ENV.DOWEL.1;
};
```

The process of interpreting the visual scene sufficiently to generate a VMR is very involved. And even when it is accomplished, this still does not yet fully ground an object or event. That is done when the agent incorporates the VMRs into memory, which is the process to which we now turn.

7.7.3 Anchoring the TMRs Associated with All RefExes in Memory

The last step of reference processing is anchoring the mentions of objects and events perceived in any way in the agent's memory. Memory management is a complex issue, involving decisions such as what to store in long-term memory versus what to discard as unimportant; when to forget previously learned information, if at all, depending on how closely LEIAs are intended to emulate people; and how to merge instances of given types of events (e.g., when several human collaborators teach a robot to perform a particular procedure in similar but not identical ways). These issues are far removed from NLU per se, and we will not discuss them further here.

To recap, during Situational Reasoning, various aspects of reference resolution are addressed: previously posited coreference links are semantically checked; some as-yet ungrounded referring expressions are grounded; and referring expressions are stored in memory if the agent decides to do so.

7.8 Residual Hidden Meanings

The question "Is there a deeper meaning?" is one that LEIAs need to consider but should not pursue too actively, as they could quickly drive their human partners crazy trying to do something about every utterance. Humans think aloud, complain, and engage in phatic exchanges without intending them to be acted on. Utterances often contain no hidden (underlying, implied) meanings, and even if they do, people often miss them. This is made clear by the frequency of such clarifications as "I was actually asking you to help me," "Are you being sarcastic?", "Does she *really* drink twenty cups of tea a day?", and "Come on, I was only joking."

As regards associated linguistic phenomena, we have made a start on modeling the detection of noncanonical indirect speech acts, sarcasm, and hyperbole, but we have not yet ventured into humor.

As-yet undetected indirect speech acts. As we saw in section 4.4, most indirect speech acts are conventionalized and can be detected using constructions recorded in the

lexicon: for example, "We need to X," and "I can't do this by myself!" However, some indirect speech acts cannot be captured by lexical senses because there are no invariable words to anchor them in the lexicon. For example, an NP fragment in isolation (i.e., not in a paired discourse pattern, like a question followed by an answer) often means *Give me NP*, as long as the object in question can, in fact, be given to the speaker by the interlocutor. This last check is sufficient to exclude an indirect speech act reading for utterances like "Nuts!" in any context that does not involve either a machine shop (where *nuts* pair with *bolts*) or eating. So, if someone says, "Chair back!" to our chair-building LEIA, it can hypothesize that the user wants to be given a chair back and can see whether that is within its capabilities (it is). We discussed the treatment of bare NPs in section 4.3.4.

Another generalization is that expressing a negative state of affairs can be a request to improve it. Acting on this generalization, however, requires understanding which states of affairs are bad, which are good, and what LEIAs can do to repair the bad ones. For example, if a person tells a furniture-building LEIA, "This nail is too short," then he or she probably wants to be given not just any longer one but one that is the next size longer, if there is such an option. This generalization applies to any multivalued objects that a LEIA can give to its collaborator: one can have large and small hammers, long and short nails, heavyweight and lightweight clamps, and so on. By contrast, if the only chair back that is available is too heavy, then the human needs to figure out what to do about it. In short, much of the reasoning involved is both domain- and task-dependent, and we will approach compiling an inventory of appropriate reasoning rules in bottom-up fashion.

Sarcasm. Although detecting sarcasm might seem like an unnecessary flourish for LEIAs, it can actually have practical importance. As we discuss in chapter 8, mindreading is an important aspect of human communication. People (largely subconsciously) construct models of each other and make decisions based on those models. It actually matters whether "I love mowing the lawn!" means that I really do love it (and, therefore, don't get in the way of my fun) or that I don't love it (and if you don't do it next week I'll be mad). One way of preparing LEIAs to detect at least some realizations of sarcasm is to describe events and states in the ontology as typically desirable or undesirable—which is a default that can, of course, be overridden for a particular individual by concrete information stored to memory.

Hyperbole. People exaggerate all the time. (Get it?) *Grandma drinks twenty cups of tea a day. If you go one-half-mile-an-hour over the speed limit on that street, they'll give you a ticket*. In terms of formal meaning representations, hyperbole is best captured by converting the stated numbers into their respective abstract representations. For our examples, this correlates with *drinking a very large amount of tea* and *going very slightly over the speed limit*.

LEIAs detect exaggerations by comparing the stated value with expectations stored in the ontology—to the extent that the needed information is available. For example, if the ontology says that people are generally not more than seven feet tall, then saying that someone is twenty feet tall is surely an exaggeration. However, although our current ontology

includes typical heights of people, it does not cover every type of knowledge, such as normal daily beverage consumption or the minimal speed infraction for getting a ticket. This need for significantly deep knowledge is why we postpone hyperbole detection until the stage of script-based reasoning. If our furniture-building robot is told, "We have to build this chair in two minutes flat!" but the LEIA's script says that the average building time is two hours, then it must interpret this as *very fast*. So, the basic TMR, generated during Basic Semantic Analysis, will include a duration of two minutes, but it can be modified at this stage to convert *two minutes* into the highest value on the abstract scale of SPEED.

As concerns detecting hyperbole, over time, the agent's TMR repository can be useful for this. Imagine that a LEIA encounters the example about Grandma drinking twenty cups of tea a day and has no way of knowing that it is an exaggeration. So the TMR states, literally, that Grandma drinks twenty cups of tea a day. However, a developer might review this TMR (which is always an option), recognize the hyperbole, and change the representation to an abstract indication of quantity—namely QUANTITY 1, which is the highest value on the abstract scale {0,1}. (The lack of a measuring unit is the clue that this is an abstract value.) The agent now has the combination of (a) the original input, (b) its mistaken analysis, and (c) the corrected analysis. This provides the prerequisites for it to reason that if Michael is said to drink twenty Cokes a day, this, too, is an exaggeration. Of course, there is nothing simple about language or the world: after all, it might be fine for a marathon runner training in a hot climate to drink twenty cups of water a day.

7.9 Learning by Reading

Learning by reading is an extension of the new-word learning the agent undertakes during Basic Semantic Analysis. There, the LEIA analyzes the meanings of new words using the semantic dependency structure and knowledge recorded in the ontology. This typically results in a coarse-grained analysis. For example, from the input *Jack is eating a kumquat*, the agent can learn only that *kumquat* is a FOOD—which is a good start, but only a start. To supplement this analysis, the agent can explore text corpora, identifying and processing sentences that contain information about kumquats. Typically, as with *kumquat*, the word has more than one sense—in this case, the word can refer to a tree or its fruit. So the agent needs to first cluster the sentences containing the word (using knowledge-based or statistical methods) into different senses and then attempt to learn the syntactic and semantic features of those senses. It can either link new word senses to the most applicable available concept in the ontology or posit a new concept in the most appropriate position in the ontological graph. What is learned can be used in various ways: it can improve a runtime analysis, it can serve as an intermediate result for semiautomatic knowledge acquisition (in order not to corrupt the quality of the knowledge bases), or it can directly modify the knowledge bases, provided that a necessary threshold of confidence is achieved.

Learning by reading has long been understood as a cornerstone of AI, since it will allow agents to convert large volumes of text into interpreted knowledge that is useful for reasoning. However, it is also among the most difficult problems, as evidenced by past experimentation both within our group (e.g., English & Nirenburg, 2007, 2010) and outside of it (e.g., Barker et al., 2007). At this point in the evolution of our program of NLU, we are preparing agents to learn by identifying and algorithmically accounting for eventualities. High-quality learning is, however, directly dependent on the size and quality of the knowledge bases (especially, lexicon and ontology) that are used to bootstrap the process. No doubt, more manual knowledge engineering is needed before we can expect agents to excel at supplementing those knowledge bases automatically.