Linguistics for the Age of AI

OA Funding Provided By:

The open access edition of this book was made possible by generous funding from Arcadia—a charitable fund of Lisbet Rausing and Peter Baldwin.

The title-level DOI for this work is:

doi:10.7551/mitpress/13618.001.0001

2

A Brief Overview of Natural Language Understanding by LEIAs

This chapter presents a brief overview of natural language understanding (NLU) by LEIAs. Our purpose is to use simple examples to describe and motivate our overall approach before introducing, in chapters 3–7, the large number of linguistic phenomena that must be treated by any realistic-scale NLU system.

One word of framing before we begin. We cannot emphasize enough that this book describes an ongoing, long-term, broad-scope program of work that we call Linguistics for the Age of AI. The depth and breadth of work to be done is commensurate with the loftiness of the goal: enabling machines to use language with humanlike proficiency. The main contributions of the book are the computational cognitive models we present and their organization into an overall, broadly encompassing process of NLU. At present, the models are of various statuses: many have been demonstrated in prototype systems, some have been formally evaluated, and still others await implementation. When we say that an agent *does X*, this describes how our model works; it does not mean that, today, our agent systems can *do X* with respect to every possible language interaction. Were that the case, we would have already solved the language problem in AI.

As the book progresses, readers should find sufficient details about a sufficient number of models to realize that we do not underestimate the sheer quantity of work awaiting linguists who take on the challenge of automating human-level NLU. Moreover, with each cycle of building, implementing, and evaluating models, new theoretical and methodological insights will accrue, which could lead to significant modifications to the approach presented here. But still, as long as the goal of the research enterprise is to build psychologically plausible, explanatory models of language behavior, we expect the core of our approach to endure.

2.1 Theory, Methodology, and Strategy

We begin with an overview of theoretical principles, methodological preferences, and strategic choices. The presentation is organized into four categories: the nature of NLU, the knowledge and reasoning needed for NLU, how NLU interacts with overall agent cognition, and strategic preferences. The lists are not exhaustive; they are intended to serve as a conceptual scaffolding for upcoming discussions.

The nature of NLU

- Language understanding by LEIAs involves translating language inputs into ontologically grounded *text meaning representations* (TMRs), which are then stored in agent memory (Nirenburg & Raskin, 2004).
- Translation into the ontologically grounded metalanguage
  - focuses on the content of the message rather than its form;
  - resolves complexities such as lexical and referential ambiguity, ellipsis, and linguistic paraphrase; and
  - permits many of the same knowledge bases and reasoning engines to be used for processing texts in different languages.
- The global interpretation of text meaning is built up compositionally from the interpretations of progressively larger groups of words and phrases.
- Semantic imprecision is recognized as a feature of natural language; it is concretized only if the imprecision impedes reasoning or decision-making.
- Semantic analysis is carried out in stages, marked by the kind of *wait and see* tactic that people seem to use when they are operating in a foreign language that they know only moderately well (i.e., even if something is not immediately clear, what comes next might serve to clarify).
- Each meaning interpretation is assigned a confidence level that reflects the degree to which the interpretation deviates from the expectations of the supporting knowledge bases and algorithms. In human-agent applications, confidence levels will help agents to decide whether or not to act on their understanding of a language input or seek clarification from a human collaborator.
- We make no theoretical claims about how people carry out pre-semantic aspects of language analysis—most notably, preprocessing and syntactic analysis; our research interests are squarely in the realm of semantics and pragmatics. Therefore, pre-semantic analysis in our environment is outsourced to externally developed engines, and the agent interprets their results as overridable heuristic evidence.

The knowledge and reasoning needed for NLU

- Language understanding relies primarily on three knowledge resources: the ontology (knowledge about concept types), episodic memory (knowledge about concept instances), and an ontological-semantic lexicon.
- Language understanding employs algorithms that manipulate both linguistic and extralinguistic knowledge.

- The microtheories for the treatment of language phenomena (e.g., lexical disambiguation, coreference resolution, indirect speech act interpretation) reflect an attempt to operationalize our understanding of how humans make sense of language. Machine learning–based mainstream NLP does not adhere to this objective. As a result, it cannot claim insights into explanatory theories of language use. Human-inspired microtheories not only have the best potential to enable agents to successfully collaborate with people but also make the results explainable in human terms.
- We consider *listing* a good and necessary approach to capturing human knowledge. This includes writing rules, lexicon entries, grammatical constructions, and more. The common prejudice against listing derives from the desire to create elegant and streamlined accounts that strike some as scientifically satisfying. However, this prioritization of the elegant is misplaced when the domain of inquiry is natural language, and it invariably leads to the proliferation of wastebasket phenomena that are not treated at all. There is no evidence that people lack the capacity to record a lot of language-oriented information directly and explicitly. Moreover, there is no reason why the principle of dynamic programming (whereby the results of computations are remembered and reused instead of being recomputed from scratch every time they are needed) should not be used in modeling human language processing.

How NLU interacts with overall agent cognition

- Agents use stored knowledge recorded as TMRs (not natural language text strings!) to reason about action. (The source text strings are, however, stored as metadata associated with each TMR since they can, e.g., inform word selection when generating a response to an input.)
- Agents are, at base, language independent. Only the text-to-TMR translation process is language specific—and even many aspects of *that* are applicable crosslinguistically. To cite just two examples, the semantic descriptions in the LEIA's lexicon are readily portable across lexicons for different languages, and key aspects of reasoning about coreference can be carried out over TMRs, not the language strings that gave rise to them.
- Language understanding cannot be separated from overall agent cognition since heuristics that support language understanding draw from (among other things) the results of processing other modes of perception (such as vision), reasoning about the speaker's plans and goals, and reasoning about how much effort to expend on understanding difficult inputs.
- Agents, like humans, must be able to leverage many reasoning strategies, including language-centric reasoning (using selectional constraints, linguistic rules, and so on), reasoning by analogy, goal-based reasoning, and statistical likelihood.

- Modeling human-level language understanding does not mean modeling what an ideal human might ideally understand given ideal circumstances. People don't operate in a perfect, sanitized world. Instead, they pay various degrees of attention to language inputs depending on whether the content is interesting and/or relevant to them. So, too, must agents.
- One way to model humanlike language understanding is to focus on *actionability*—that is, configuring agents to pursue a level of interpretation that supports intelligently selected subsequent action. An actionable interpretation might represent a complete and correct analysis of an input, or it might be incomplete; it might involve deep analysis or only skimming the surface; and it might be achievable by the agent alone, or it might require clarifications or corrections by a human or artificial collaborator. Deeming an interpretation actionable serves as a practical halting condition for language analysis.
- Among the available actions an agent can take in response to an input are physical actions, verbal actions, and mental actions. The latter include decision-making, learning, and even ignoring the input, having deemed it outside the agent's current topics of interest.

Strategic preferences

- Although our program of work focuses primarily on the research end of the R&D spectrum, system implementations serve to validate the component microtheories and algorithms. Implementations are both theoretically necessary (AI requires computation) and strategically beneficial (we always discover something unexpected when applying algorithms to unconstrained texts).
- Since knowledge bases compiled by our team in decades past offer sufficient breadth and depth of coverage to serve as test beds for developing theories and models, manual resource-building is not a current priority, even though continued expansion of the knowledge resources is essential for achieving human-level AI on an industrial scale.
- A scientifically more compelling solution to the need for knowledge is making the agent capable of lifelong learning by reading, by experience, and by being told—which is something we *are* working on.
- It is essential to distinguish domain-neutral from domain-specific facets of language understanding and to design NLU systems that are maximally portable across domains. This counters the implicit, but unrealistic, hope of developers of narrow-domain systems: that those systems will ultimately converge, resulting in open-domain coverage. This will not happen because narrow-domain systems do not address the core problem of lexical ambiguity that looms large when moving to the open domain.
- System development in our approach is task-oriented, not method-oriented. The task is to develop humanlike NLU capabilities that permit LEIAs to explain their

cognitive functioning in human terms. Only then will they become trusted collaborators. Any method that is well suited to a particular task can serve this goal. It so happens that knowledge-based methods are best suited to solving most NLU tasks, but when other methods prove useful, they are incorporated. This task-oriented methodology is in contrast to first choosing a method (such as using statistical algorithms operating over big data) and then seeking useful applications for it.

- The cognitive models contributing to NLU aim for descriptive adequacy, not neatness. Decision-making during modeling is guided by the principle *choose and move on*. A sure route to failure is to ponder over innumerable decision spaces with their associated pros and cons, effectively paralyzing the enterprise with the drive for perfect but nonexistent solutions. The *choose and move on* principle reflects a preference for something over nothing, as well as the reality that all models are incomplete (Bailer-Jones, 2009).
- Strategic simplifications are incorporated, provided they do not jeopardize the utility of the results. If a simplification causes a drop in quality, the optimal grain size of description must be reconsidered. For example, to date we have not worked on applications for which distinctions between types of dogs, hats, or lizards were important. Accordingly, all names of dog breeds, types of hats, and infraorders of lizards are mapped to the ontological concepts DOG, HAT, and LIZARD, respectively. However, as soon as a dog-, hat-, or lizard-oriented application comes along, then this simplification will no longer serve, and the requisite knowledge-acquisition work will come on agenda.

As the above theoretical and methodological statements should make clear, our approach to extracting meaning from natural language text is as human inspired as possible for the development of theory, but as methodologically inclusive as necessary for the development of application systems.

An organizational sidebar. This book contains many examples of knowledge structures—text meaning representations, lexical senses, ontological frames, and more. In our NLU system, these are recorded using a formalism that takes time to learn to read quickly. Although assessing the quality of formalisms is important for gauging both their expressive power and their utility for building computational applications, this is not central to the goals of this book. Here, we concentrate on the *content* of our approach to NLU, as reflected in microtheories and algorithms that process them. To keep the cognitive load for readers at a reasonable level, we decided to present knowledge structures in a simplified format, with the goal of making the specific, contextually relevant points as clear as possible. Naturally, examples retain certain features of the underlying system-oriented formalism. Readers interested in the formalism itself will find examples in the online appendixes at https://homepages.hass.rpi.edu/mcsham2/Linguistics-for-the-Age-of-AI.html.

2.2 A Warm-Up Example

To reiterate, a LEIA's understanding of what a language input means is recorded in an ontologically grounded *text meaning representation*, or TMR. Consider the simplified[1] TMR for ***A gray squirrel ate a nut.***

INGEST-1	
AGENT	SQUIRREL-1
THEME	NUT-FOODSTUFF-1
TIME	*<find-anchor-time*
from-sense	*eat-v1*
word-num	*3*
SQUIRREL-1	
COLOR	gray
AGENT-OF	INGEST-1
from-sense	*squirrel-n1*
word-num	*2*
NUT-FOODSTUFF-1	
THEME-OF	INGEST-1
from-sense	*nut-n1*
word-num	*5*

This example is simple for the following reasons: it contains just one clause; that clause is syntactically regular; none of its referring expressions require coreference resolution; its lexical ambiguities can be resolved using rather simple analysis techniques; and the semantic analyses of the lexemes reliably combine into an ontologically valid semantic dependency structure. This TMR should be read as follows.

- The first frame is headed by a numbered instance of the concept INGEST. Concepts are distinguished from words of English by the use of small caps. Note that this is not vacuous *upper-case semantics*[2] because the concepts in question have formal definitions in the ontology that (a) are based on value sets of ontological properties (the primitives in the conceptual system of the ontology) and (b) support reasoning about language and the world.
- INGEST-1 has three contextually relevant property values: its AGENT (the eater) is an instance of SQUIRREL; its THEME (what is eaten) is an instance of NUT-FOODSTUFF; and the TIME of the event is before the time of speech.[3]
- The properties in italics are among the many elements of metadata generated during processing, which support system evaluation, testing, and debugging. Those shown indicate which word number (starting with 0) and which lexical sense were used to generate the given TMR frame. In most TMRs presented hereafter, we will not include these metadata slots.

- The next frame, headed by SQUIRREL-1, shows not only the inverse relation to INGEST-1 but also that the COLOR of this SQUIRREL is gray. *Gray* is not written in small caps because it is a literal (not concept) filler of the property COLOR.
- Since we have no additional information about the nut, its frame—NUT-FOODSTUFF-1—shows only the inverse relation with INGEST-1, along with the same type of metadata described above.

For each TMR it produces, the LEIA generates a value of the *confidence* parameter (a type of metadata not shown in the example) that reflects its certainty in the TMR's correctness. For TMRs, like this one, that do not require advanced semantic and pragmatic reasoning, the confidence score is computed using a function that compares how the elements of input align with the syntactic and semantic expectations of word senses in the lexicon.

In working through how a LEIA generates this analysis, we will assume for the moment (since we're just getting warmed up) that the agent has access to the entire sentence at once. We will introduce incremental processing—our first complicating factor—in section 2.4.

First the input undergoes preprocessing and syntactic analysis, which are provided by an external toolset.[4] Using features from the syntactic parse, the LEIA attempts to align sentence constituents with the syntactic expectations recorded in the lexicon for the words in the sentence. For example, it will find three senses of the verb *eat* in the lexicon. One is optionally transitive and means INGEST. The other two describe the idiom *eat away at* in its physical sense (*This powerful antioxidant is always a handy chemical for the aerospace industry, since it can eat away at metal without causing the heat fatigue associated with traditional machining.* $_{\text{(COCA)}}$) and its abstract sense (*This vice begins to eat away at our soul …* $_{\text{(COCA)}}$).[5] Since the idiomatic senses require the words *away at*, which are not present in the input, they are rejected, leaving only the INGEST sense as a viable candidate. Below is the needed lexical sense of *eat* (eat-v1) followed by one of the idiomatic senses (eat-v2).

```
eat-v1
    def.                ingest
    ex.                 He was eating (cheese).
    syn-struc
        subject         $var1
        v               $var0
        directobject    $var2 (opt+)
    sem-struc
        INGEST
            AGENT       ^$var1
            THEME       ^$var2 (sem FOOD)

eat-v2
    def.        The construction 'eat away at': erode physically
    ex.         The rust ate away at the pipe.
    syn-struc
```

```
    subject   $var1
    v         $var0
    adv       $var2 (root 'away')
    pp
        prep   $var3 (root 'at')
        obj    $var4
sem-struc
    ERODE
        INSTRUMENT      ^$var1
        THEME           ^$var4
    ^$var2   null-sem+
    ^$var3   null-sem+
```

We will explain the format and content of lexical senses using *eat-v1* as an example. The syntactic structure (syn-struc) zone of *eat-v1* says that this sense of *eat* is optionally transitive: it requires a subject and can be used with or without a direct object (*opt+*means *optional*).

Each constituent of input is associated with a variable in the syn-struc. In *eat-v1*, the head of the entry (the verb) is $var0, the subject is $var1, and the direct object is $var2. The meaning of $var0, expressed as an ontological concept, heads the semantic structure (sem-struc) zone: INGEST. After all, the point of this sense is to describe the meaning of $var0 when it is used in this particular construction. All other variables are linked to their semantic interpretations in the sem-struc, with ^ being read as *the meaning of*. So the *word* that fills the subject slot in the syn-struc, $var1, must be semantically analyzed, resulting in ^$var1. Then ^$var1 must be evaluated to see if it is a semantically suitable AGENT of an INGEST event.

Note that, by default, the semantic constraints on the case roles are not listed because they are drawn from the ontology. However, the THEME of INGEST is an exception. Its constraint is listed because it overrides (i.e., is narrower than) what is listed in the ontology: whereas one can INGEST food, beverages, or medication (see the ontological frame for INGEST below), one can *eat* only food.[6]

Shifting for a moment to *eat-v2*, two of its variables are described in the sem-struc as *null-sem+*. This means that their meaning has already been taken care of by the semantic representation and should not be computed compositionally. In this example, *eat away at*, taken together, means ERODE; *away* and *at* do not carry any extra meaning beyond that.

The ontology, for its part, provides information about the valid fillers of the case roles of INGEST. Consider a small excerpt from the ontological description of INGEST; the full concept description contains many more property-facet-value triples.

```
INGEST
    AGENT       sem            ANIMAL
                relaxable-to   SOCIAL-OBJECT
    THEME       sem            FOOD, BEVERAGE, INGESTIBLE-MEDICATION
                relaxable-to   ANIMAL, PLANT
                not            HUMAN
```

This ontological frame says that the typical AGENT of INGEST (i.e., the basic semantic constraint indicated by the *sem* facet) is an ANIMAL; however, this constraint can be relaxed to SOCIAL-OBJECTS, as in *The fire department eats a lot of pizza*. Similarly, the description of the THEME indicates that FOOD, BEVERAGE, and INGESTIBLE-MEDICATION are the most typical THEMES, but other ANIMALS and PLANTS not already subsumed under the FOOD subtree might be consumed as well. HUMANS are explicitly excluded as ingestibles (which illustrates the semantics of the *not* facet), since they would otherwise be understood as unusual-but-possible ingestibles due to their placement in the ANIMAL subtree of the ontology. There are two important reasons to exclude humans as ingestibles even though, for sure, big cats and alligators have been known to occasionally eat a person. First, the ontology is intended to provide agents with knowledge of how the world typically works. Second, there is a sense of *eat* that means *to annoy* (*What was eating her?* $_{\text{(COCA)}}$), and that sense should be preferred when the direct object is a person.

Having narrowed down the interpretation of *eat* to a single sense, the LEIA must now determine which senses of *squirrel*, *gray*, and *nut* best fit this input. *Squirrel* and *gray* are easy: the lexicon currently contains only one sense of each, and these senses fit well semantically: SQUIRREL is a suitable AGENT of INGEST, and *gray* is a valid COLOR of SQUIRREL. However, there are three senses of *nut*: an edible foodstuff, a crazy person, and a machine part. We just saw that neither people nor machine parts are FOOD, leaving only the NUT-FOODSTUFF sense, which is selected as a high-confidence interpretation.

Operationally speaking, after all the constraints have been checked, the TMR for *A gray squirrel ate a nut* is generated by

1. copying the sem-struc of *eat-v1* into the nascent TMR;
2. translating the concept type (INGEST) into an instance (INGEST-1); and
3. replacing the variables with their appropriate interpretations: ^$var1 becomes SQUIRREL-1 (COLOR gray), and ^$var2 becomes NUT-FOODSTUFF-1.

With respect to runtime reasoning, this example is as straightforward as it gets since (a) it involves only matching statically recorded constraints, and (b) all constraints match in a unique and satisfactory way. *Straightforward constraint matching* does not, however, come for free: its precondition is the availability of high-quality lexical and ontological knowledge bases that are sufficiently detailed to allow the LEIA to disambiguate words and validate the semantic congruity of its resulting interpretations.

As mentioned earlier, LEIAs generate confidence scores for particular TMRs based on how well the syntactic and semantic expectations of lexical senses are satisfied by the candidate interpretation. Whereas our example TMR will get a very high confidence score, there will be no high-scoring interpretations for *That furry face is eating a nut*. Such inputs are handled using recovery methods described in later chapters.

The ontologically grounded knowledge representation language just illustrated has many advantages for agent reasoning (McShane & Nirenburg, 2012). Most importantly, (a) it is unambiguous, and (b) the concepts underlying word senses are described extensively in the ontology, which means that more knowledge is available for reasoning about language and the world than is made available by the occurrence of words in the input. However, translating natural language utterances into this metalanguage is difficult and expensive. So, a reasonable question is, *Do we really need it?*

If agents were to communicate exclusively with other agents, and if they had no need to learn anything from human-oriented language resources, then there would be no need for the natural-language-to-knowledge-representation-language translation that we describe. However, for agents to be truly useful, they *do* need to communicate with people, and they *do* need to learn about the world by converting vast amounts of data into interpreted knowledge. Because of this, it is important to both establish the formal relationship between natural language and a knowledge representation language and to provide intelligent agents with the facility to translate between them. For other views on the relationship between natural language and knowledge representation languages, see the deep dive in section 2.8.1.

2.3 Knowledge Bases

The main static knowledge bases for LEIAs are

1. the lexicon (including an onomasticon—a repository of proper names), which describes the syntactic expectations of words and phrases along with their ontologically grounded meanings;
2. the ontology, which is the repository of types of objects and events, each of which is described by a large number of properties; and
3. the episodic memory, which is the repository of concept instances—that is, real-life representatives of objects and events, along with their property values.[7]

Theoretically speaking, every LEIA—like every person—will have idiosyncratic knowledge bases reflecting their individual knowledge, beliefs, and experiences. And, in fact, such individualization does occur in practice since LEIAs not only remember the new information they learn through language understanding but can also use this information to dynamically learn new words and ontological concepts in various ways (see chapter 8). However, this learning builds on the core lexicon, ontology, and episodic memory that are provided to all LEIAs as a model of a typical adult's knowledge about language and the world. The coverage of these core knowledge bases is, of course, incomplete relative to the knowledge store of an average person (a practical matter); but, for what they cover, they are representative. For example, the word *horse* has three senses in the current lexicon, referring to an animal, a piece of gymnastic equipment, and a sawhorse. So every time a LEIA encounters the word *horse* it must contextually disambiguate it, which it does using ontological and contextual knowledge.

Apart from LEIAs that model typical adults, there are also specialist LEIAs that are endowed with additional ontological, lexical, and episodic knowledge in a particular domain. For example, LEIAs serving as tutors and advisors in the field of clinical medicine must not only have extensive knowledge of that domain but also be aware of the differences between their knowledge and that of a typical person. This can be operationalized by flagging specialist-only subtrees in the ontology as well as specialist-only lexicon entries. When agents use these flagged concepts, words, and phrases in dialogs with nonspecialists, they, like people, will introduce them with explanations. This kind of mindreading allows for effective communication between individuals possessing different levels of expertise (see chapter 8 for more on mindreading). After all, if a physician's ten-minute explanation of all the potential side effects of a medication is so packed with specialist terminology that the patient understands nothing, then the communication has failed.

In principle, any LEIA can have or lack any datum, and it can have wrong beliefs about things as well. This is an interesting aspect of cognitive modeling: preparing agents to behave in lifelike ways in the face of incomplete and contradictory beliefs. However, individual differentiation is not the focus of the current discussion. Here, we concentrate on the basics: the knowledge bases that reflect general adult-level knowledge about language and the world.

The sections that follow give a very brief, nontechnical introduction to these knowledge bases. Readers interested in more detail can consult the cited references.

2.3.1 The Ontology

The LEIA's ontology is a formal model of the world that is encoded in the metalanguage presented above. A comprehensive description of, and rationale for, the form and content of the ontology is available in Nirenburg and Raskin (2004, section 7.1). Here we present just enough detail to ground the description of NLU to come.

The ontology is organized as a multiple-inheritance hierarchical collection of frames headed by concepts that are named using language-independent labels. It currently contains approximately 9,000 concepts, most of which belong to the general domain. We avoid a proliferation of ontological concepts, in line with the recommendation by Hayes (1979) that the ratio of knowledge elements used to describe a set of elements of the world to the number of these latter elements must be kept as low as possible. There are additional reasons why the number of concepts in the ontology is far lower than the number of words or phrases in any language.

1. Synonyms map to the same ontological concept, with semantic nuances of particular words recorded as constraints in the corresponding lexical senses.
2. Many lexical items are interpreted using combinations of concepts.
3. Lexical items that represent a real or abstract point or range on a scale all point to a single property that represents that scale (e.g., *brilliant*, *smart*, *pretty smart*, and *dumb* reflect different values of INTELLIGENCE).

4. Concepts are intended to be crosslinguistically and cross-culturally relevant. This means that the ontology does not, for example, contain a concept for the notion *recall* in the sense "request that a purchased good be returned because of a discovered flaw" because not all languages and cultures use this concept. Instead, the meanings of such words are described compositionally in the lexicons of those languages that do use them.

Concepts divide up into EVENTs, OBJECTs, and PROPERTYs. PROPERTYs are primitives, which means that their meaning is grounded in the real world with no further ontological decomposition. Ontological properties are used to define the meaning of OBJECTs and EVENTs. Stated plainly, an OBJECT or EVENT *means* whatever its property-facet-value triples say it means. The types of properties contributing to OBJECT and EVENT descriptions include:

- IS-A and SUBCLASSES, which are the two properties that indicate the concept's placement in the tree of inheritance. Multiple inheritance is permitted but not overused, and rarely does a concept have more than two parents.
- RELATIONs, which indicate relationships among OBJECTs and EVENTs. Examples include the nine case roles that describe the typical participants in EVENTs—AGENT, THEME, BENEFICIARY, EXPERIENCER, INSTRUMENT, PATH, SOURCE, DESTINATION, LOCATION—along with their inverses (e.g., AGENT-OF, THEME-OF).[8]
- SCALAR-ATTRIBUTEs, which indicate meanings that can be expressed by numbers or ranges of numbers: for example, COST, CARDINALITY, FREQUENCY.
- LITERAL-ATTRIBUTEs, which indicate meanings whose fillers were determined by acquirers to be best represented by uninterpreted literals: for example, the property MARITAL-STATUS has the literal fillers *single*, *married*, *divorced*, *widowed*.
- Several administrative properties for the use of people only, such as DEFINITION and NOTES.

Selecting the optimal inventory of properties has not been, nor is it slated to be, on agenda in our research—though it is an interesting topic for full-time ontologists. Instead, we have taken a practical, system-oriented approach to creating properties. Some were included by virtue of being central to any world model: for example, HAS-CAPITAL is a useful descriptor for LARGE-GEOPOLITICAL-ENTITYs, and CAUSED-BY is needed to describe EVENTs. Other properties are convenient shorthand, introduced for a given application: for example, HAS-COACH was included for an Olympics application because it was more convenient to record and manipulate structures like "X HAS-COACH Y" than more explanatory structures like "there is a long-term, repeating succession of COACHING-EVENTS for which Y is the AGENT and X is the BENEFICIARY." (For more on semantically decomposable properties, see section 6.1.3.) The point is that practically any inventory of properties can serve a LEIA's purposes as long as those properties are used effectively in ontological descriptions of related OBJECTS and EVENTS, and as long as the LEIA's reasoners are configured to appropriately use them.

The expressive power of the ontology is enhanced by multivalued fillers for properties, implemented using *facets*. Facets permit the ontology to include information such as *the most typical colors of a car are white, black, silver, and gray; other normal, but less common, colors are red, blue, brown, and yellow; rare colors are gold and purple*. The inventory of facets includes: *default*, which represents the most restricted, highly typical subset of fillers; *sem*, which represents typical selectional restrictions; *relaxable-to*, which represents what is, in principle, possible although not typical; and *value*, which represents not a constraint but an actual, nonoverridable value. *Value* is used primarily in episodic memory, but in the ontology it has the role of indicating the place of the concept in the hierarchy, using the concepts IS-A and SUBCLASSES. Select properties from the ontological frame for the event DRUG-DEALING illustrate the use of facets.

DRUG-DEALING

IS-A	value	CRIMINAL-ACTIVITY
AGENT	default	CRIMINAL, DRUG-CARTEL
	sem	HUMAN
	relaxable-to	SOCIAL-OBJECT
THEME	default	ILLEGAL-DRUG
INSTRUMENT	sem	MONEY
HAS-EVENT-AS-PART	sem	BUY, SELL
LOCATION	default	CITY
	sem	PLACE
	relaxable-to	PHYSICAL-OBJECT
…		

OBJECTS and EVENTS are defined in the ontology using an average of sixteen properties each, but many of the fillers of those properties are inherited rather than locally specified.

To reiterate the most important point: The *meaning* of an OBJECT or EVENT is the set of its property-facet-value triples.

The main benefits of writing an ontology in a knowledge representation language rather than a natural language are (a) the absence of ambiguity in the knowledge representation language, which makes the knowledge suitable for automatic reasoning, and (b) its reusability across natural languages. Cut to thirty years from now, and the LEIA's ontology should contain tens of thousands of well-described concepts, including thousands of descriptions of complex events (scripts). Since the ontology is language independent, this knowledge infrastructure will be accessible to intelligent agents that communicate in any language, as long as a compatible lexicon and language-understanding engine for that language have been developed.

A core need in ontological modeling is describing complex events that involve multiple steps, multiple participants, and multiple props. In our ontology these complex events are represented using ontological scripts.[9] Scripts can reflect knowledge in any domain (what happens at a doctor's appointment, how to make spaghetti and meatballs, how to remove a brain tumor), and they can be at any level of generality (from a basic sequence of events to the level of detail needed to generate computer simulations). What ontological scripts do *not* do is conform to the simple slot-facet-filler formalism described above. Although scripts use the same concepts and the same basic knowledge representation language, they require additional expressive power. Taking examples from the domain of medical appointments, scripts require:

- The coreferencing of arguments. In a given appointment, the same instance of PHYSICIAN will carry out many actions (e.g., asking questions, answering questions, recommending interventions), and the same instance of MEDICAL-PATIENT will carry out many actions (asking questions, answering questions, deciding about interventions).
- Loops. There can be many instances of event sequences, such as ask/answer a question and propose/discuss an intervention.
- Variations in ordering. A doctor can get vital signs before or after the patient interview, and provide lifestyle recommendations before or after discussing medical interventions.
- Optional components. A doctor may or may not engage in small talk and may or may not recommend tests or interventions.
- Time management. For simulation-oriented or time-sensitive scripts (e.g., in the domain of emergency medicine), the script must include information about what happens when, how fast, and for how long.

In short, although scripts *are* a part of ontology per se—that is, they fill the HAS-EVENT-AS-PART slot of the ontological frames for complex EVENTS—they are not simple slot-filler knowledge of the type illustrated earlier.

So, what does a script look like? The simplest way to answer is by example. Below is a tiny excerpt—in its original, unsimplified, format—from the script that supports the interactive simulation of virtual patients experiencing gastroesophageal reflux disease (GERD) in the Maryland Virtual Patient (MVP) application of LEIAs (see chapter 8 for further discussion).

```
(tracks
    (if (>total-time-in-acid-reflux 1.2)
        Then
        (bind-variables
            (fraction-through (- 1.0 (/ (- time-since-start time-increment) gerd-time)))
            (gerd-t-duration-days (get-attribute gerd-t-duration gerd-patient 1000))
            (extra-time (fraction-through gerd-t-duration-days 60 60 24)))
        (effect
```

```
        (unassert-background-script heal-preclinical-gerd $var0)
        (assert-background-script preclinical-gerd $var0
            ((extra-time (+extra-time time-increment))))))
(if (and    (>=time-since-start gerd-time)
            (<=total-time-in-acid-reflux 1.2))
    Then
    (effect
        (unassert-background-script heal-preclinical-gerd $var0))))
```

The complication with presenting this example is that this script illustrates difficult issues of dynamic (in this case, physiological) simulation, including time management, feature-value checking and updating, cause-effect relationships, and the assertion and unassertion of interrelated scripts. Rather than simplifying the format, as we do with other structures we illustrate, it was easier to present it in its internal Lisp form and accept that it reveals as much about engineering as it does about the underlying ontological knowledge.

Not all scripts must support dynamic simulations. There are also more familiar workflow scripts that describe, for example, how a doctor should go about diagnosing a patient. These were used by the mentoring agent in the MVP application, whose task was to watch the actions of the user (who played the role of attending physician) and determine whether they conformed to good clinical practice. The same formalism is used for representing both of the above kinds of scripts.[10]

Scripts are used to support agent reasoning, including reasoning about language. Since this book is centrally about language, it is this kind of reasoning support that we are interested in. The following example illustrates the use of scripts in language-oriented reasoning. Consider the interaction: *"How was your doctor's appointment?" "Great! The scale was broken!"* Why does the second speaker say *the scale*? What licenses the use of *the*, considering that this object was not previously introduced into the discourse? The mention of a doctor's appointment prepares the listener to mentally access objects (like *scale*) and events that are typically associated with a doctor's appointment, making those objects and events primed for inclusion in the situation model (discourse context). In fact, the linguistic licensing of *the* with *scale* is evidence that such script activation actually takes place. Of course, it is our script-based knowledge that also explains why the person is happy, and it further allows us to infer the body type of the speaker. If we want LEIAs to be able to reason at this level as well, then scripts are the place to store the associated knowledge.

For further discussion of issues related to the content and acquisition of ontology, see the deep dive in section 2.8.2.

2.3.2 The Lexicon

A LEIA's lexicon for any language maps the words, phrases, and constructions of the language to the concepts in the ontology. The defining features of the current English lexicon are as follows:

- It contains both syntactic and semantic descriptions, linked by variables.
- It contains around 30,000 word senses.
- Open-domain vocabulary is covered, with some areas of specialization reflecting past application areas.
- Most semantic descriptions express meaning using ontological concepts, either by directly mapping to a concept or by mapping to a concept and then modifying it using property-based constraints. However, the meaning of some words, like the adverb *respectively,* must be dynamically computed in each context. In such cases, the lexical description includes a call to a procedure to carry out the necessary computation.
- During lexical acquisition, acquirers attempt to include sufficient constraints to enable the system to disambiguate the words of input at runtime. It would make no sense for a computational lexicon to split senses as finely as many human-oriented lexicons do if the agent has no way of choosing between them.
- To date, we have made it a priority to acquire frequent and semantically complex argument-taking words, such as *have* and *make*, because preparing agents to treat those hard cases is at the core of the scientific work. Acquiring a large number of nouns, such as kinds of birds or trees, is much simpler and could be done by less-trained individuals (as resources permit)—and even automatically by the agent itself through learning by reading.
- The lexicon covers all parts of speech: noun, verb, adjective, adverb, conjunction, article, quantifier, relative pronoun, number, pronoun, reflexive pronoun, auxiliary.
- It accommodates multiword expressions and constructions of any structure and complexity, as described in section 4.3.

Although we already briefly introduced the lexicon, its role in NLU is so important that we will present a few more example entries to reinforce the main points. Note that in these lexicon examples, we make explicit the ontological constraints on case roles that are drawn from the ontology, as they are accessible to the system when it processes inputs, and these are key to understanding how automatic disambiguation works. The first example juxtaposes two verbal senses of *address*.

```
address-v1
    def.    to talk to—usually, formally
    ex.     He addressed the audience.
    syn-struc
        subject         $var1
        v               $var0
        directobject    $var2
    sem-struc
        SPEECH-ACT
            AGENT           ^$var1 (sem HUMAN)
            BENEFICIARY     ^$var2 (sem HUMAN) (relaxable-to ANIMAL)
```

address-v3
  def. to consider, think about
  ex. He addressed the problem.
  syn-struc
    subject $var1
    v $var0
    directobject $var2
  sem-struc
    CONSIDER
      AGENT ^$var1 (sem HUMAN)
      THEME ^$var2 (sem ABSTRACT-OBJECT)

Syntactically (as shown in the syn-struc zones), both senses expect a subject and a direct object in the active voice, filled by the variables $var1 and $var2, respectively. However, the meanings of the direct objects are constrained differently, as shown in the respective sem-strucs. In address-v1 the meaning of the direct object (^$var2) is constrained to a HUMAN or, less commonly, an ANIMAL, whereas in address-v3 the meaning of the direct object is constrained to an ABSTRACT-OBJECT. This difference in constraints permits the analyzer to disambiguate. If the direct object in an input sentence is abstract, as in *He addressed the problem*, then *address* will be analyzed as an instance of the concept CONSIDER using address-v3. By contrast, if the direct object is human, as in *He addressed the audience*, then *address* will be analyzed as SPEECH-ACT using address-v1. The semantic roles that each variable fills are explicitly indicated in the sem-struc zone as well. In both of the senses presented here, the meaning of $var1 (^$var1) fills the AGENT role. In address-v1, the meaning of $var2 (^$var2) fills the BENEFICIARY role, whereas in address-v3, the meaning of $var2 (^$var2) fills the THEME role.

The examples above illustrate how lexically recorded *semantic* constraints support disambiguation, given the same syntactic structure. However, *syntactic* constraints can also support disambiguation. Consider the four senses of *see* shown below. The latter two require, respectively, an imperative construction (see-v3) and a transitive construction that includes a PP headed by *to* (see-v4). These syntactic constraints, along with the associated semantic constraints, provide strong heuristic guidance for automatic disambiguation.

see-v1
  def. to perceive visually
  ex. He saw her new car.
  syn-struc
    subject $var1
    v $var0
    directobject $var2
  sem-struc
    INVOLUNTARY-VISUAL-EVENT
      AGENT ^$var1 (sem ANIMAL)
      THEME ^$var2 (sem PHYSICAL-OBJECT)

```
see-v2
    def.    to consult with for advice
    ex.     Grandma saw her doctor.
    syn-struc
        subject         $var1
        v               $var0
        directobject    $var2
    sem-struc
        PROFESSIONAL-CONSULTATION
            AGENT           ^$var2 (sem MEDICAL-ROLE, LEGAL-ROLE)
            BENEFICIARY     ^$var1 (sem HUMAN)

see-v3
    def.    to refer to a portion of text
    ex.     For details, see chapter 3.
    syn-struc
        v               $var0 (form imperative)
        directobject    $var1
    sem-struc
        READ
            THEME       ^$var1 (sem TEXT-UNIT)

see-v4
    def.    to accompany someone somewhere
    ex.     He saw me to my car.
    syn-struc
        subject         $var1
        v               $var0
        directobject    $var2
        pp
            prep        $var3 (root 'to')
            obj         $var4
    sem-struc
        ESCORT
            AGENT           ^$var1 (sem HUMAN)
            BENEFICIARY     ^$var2 (sem HUMAN)
            DESTINATION     ^$var4 (sem PLACE) (relaxable-to PHYSICAL-OBJECT)
        ^$var3      null-sem+
```

A global rule used for disambiguation is to prefer analyses that fulfill more specific (i.e., narrower) constraints. In most cases, this rule works well: after all, when one says *I saw my doctor yesterday*, it typically refers to PROFESSIONAL-CONSULTATION—unless, of course, one adds the adjunct *at a basketball game*, in which case INVOLUNTARY-VISUAL-EVENT is the best choice. As people, we make the latter adjustment based on the knowledge that one consults with physicians in medical buildings, not at basketball games. While such knowledge about where events typically occur is recorded in the LEIA's ontology, we are still working toward compiling a sufficient inventory of reasoning rules to exploit it.

2.3.3 Episodic Memory

Episodic memory records the agent's knowledge of instances of objects and events. This knowledge can result from language understanding, vision processing, the interpretation of stimuli generated by computer simulations, the agent's recording of its own actions, and its memories of its own reasoning and decision-making. Entries in episodic memory are essentially TMRs with some additional metadata. In this book we will not discuss episodic memory in detail, but it is worth noting that managing it involves many important issues of cognitive modeling, such as consolidating information about objects and events presented at different times or by different sources, creating generalizations from repeating events, and modeling forgetting. In NLU, episodic memory is needed to support reference resolution (section 7.7), analogical reasoning (section 6.1.6), and learning (section 8.3). In discussing these capabilities, we will assume that the agent's episodic memory is, essentially, a list of remembered TMRs.

2.4 Incrementality

The original formulation of Ontological Semantics (Nirenburg & Raskin, 2004) was oriented around processing inputs as full sentences.[11] That was natural at the time: the text genre our team concentrated on was formal (typically journalistic) prose; syntactic parsers worked at the level of full sentences; and the applications being served were not time-sensitive. However, over the past fifteen years we realized that the best way for computationally oriented linguists to contribute to AI is in the area of human-AI collaboration. As a result, our research interests have shifted to agent applications, the genre of dialog, and modeling strategies that will allow agents to use language in maximally humanlike ways.

Human language understanding is incremental, as evidenced by behaviors such as finishing other people's sentences, interrupting midsentence to ask for clarification, and undertaking action before an utterance is complete (*Pass me that spatula and …* [the interlocutor should already be in the process of spatula passing]).[12] Accordingly, LEIAs need to be able to process language incrementally *if this will best serve their goals*. This condition is important: the additional processing demands imposed by incrementality are not always needed, and it would be unwise to make LEIAs always process language incrementally simply because they can. Incrementality is just one of the many tools in a LEIA's NLU toolbox, to be used, as warranted, to optimize its functioning.

Consider the incremental analysis of the input *Audrey killed the motor*, presented, for reasons of clarity, with only a subset of details. The first word of input is *Audrey*. The system's onomasticon contains only one sense of this string, so the nascent TMR is

```
HUMAN-1
   GENDER               female
   HAS-PERSONAL-NAME    'Audrey'
```

The next word is *killed*, so the combination *Audrey killed* is analyzed. The lexicon currently has five senses of *kill*, but only three of them permit a HUMAN to fill the subject slot:

1. cause to die: *Who do you think killed the guard?* (COCA)
2. cause to cease operating: *She slowed the ATV to a halt and killed the engine.* (COCA)
3. thwart passage of, veto: *You've killed the legislation on tobacco.* (COCA)

The other two senses can be excluded outright since one requires the subject to be an event (*When I was 10 my father died—he was a miner and lung disease killed him.* (COCA)) and the other requires it to be a nonhuman object that serves as an instrument (*The bomb killed the guy next to him …* (COCA)). The fragment *Audrey killed* offers the three equally acceptable TMR candidates, as follows:

TMR candidate 1 for *Audrey killed*

KILL-1
  AGENT HUMAN-1
  TIME <*find-anchor-time*

HUMAN-1
  GENDER female
  HAS-PERSONAL-NAME 'Audrey'

TMR candidate 2 for *Audrey killed*

VETO-1
  AGENT HUMAN-1
  TIME <*find-anchor-time*

HUMAN-1
  GENDER female
  HAS-PERSONAL-NAME 'Audrey'

TMR candidate 3 for *Audrey killed*

ASPECT-1
  PHASE end
  SCOPE OPERATE-DEVICE-1

OPERATE-DEVICE-1
  AGENT HUMAN-1
  TIME <*find-anchor-time*

HUMAN-1
  GENDER female
  HAS-PERSONAL-NAME 'Audrey'

The next word of input is *the*. The LEIA does not launch a new round of semantic analysis for the fragment *Audrey killed the* because no useful information can be gleaned from function words without their heads.

The next and final stage of analysis is launched on the entire sentence *Audrey killed the motor*. Each of the three still-viable lexical senses of *kill* includes semantic constraints on the direct object: for sense 1 it must be an ANIMAL; for sense 2, an ENGINE; and for sense 3, a BILL-LEGISLATIVE. Since *motor* maps to the concept ENGINE, sense 2—'cause to cease operating'—is selected, and the final TMR for *Audrey killed the motor* is

```
ASPECT-1
    PHASE          end
    SCOPE          OPERATE-DEVICE-1

OPERATE-DEVICE-1
    AGENT          HUMAN-1
    THEME          MOTOR-1
    TIME           <find-anchor-time

HUMAN-1
    GENDER                 female
    HAS-PERSONAL-NAME      'Audrey'
```

For the above illustration we chose a simple example. In reality, most sentences involve much more midstream ambiguity, resulting in many more candidate analyses. Moreover, it is not unusual for the LEIA to be unable to fully resolve all the ambiguities, even given the full sentence, when using ontological and lexical constraints alone—other contextual information can be needed. The point here is that LEIAs can narrow down analyses midstream, just like people can, as more elements of input become available.

2.5 The Stages of NLU and Associated Decision-Making

The incrementality just described can more precisely be called *horizontal* incrementality, since it involves processing words of input as they appear in the transcribed language stream. It juxtaposes with another important manifestation of incrementality: *vertical* incrementality. This refers to the depth of analysis applied to any input fragment. When a LEIA leverages more context, this can mean either processing more elements of input (horizontal incrementality) or leveraging more knowledge resources and reasoning algorithms to analyze the given elements of input (vertical incrementality). The availability of horizontal and vertical incrementality during NLU is graphically represented in figure 2.1.

As an illustration of these notions of incrementality, consider the following examples, in which underlining separates text chunks that will be consumed sequentially during incremental semantic analysis.

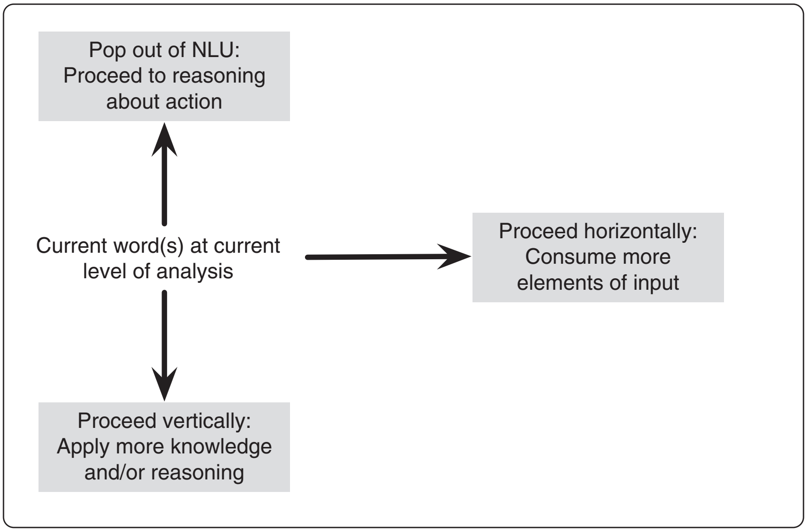

Figure 2.1
Horizontal and vertical incrementality.

(2.1) A black bear ___ is eating ___ a fish.

(2.2) My monkey ___ promised ___ he ___ wouldn't do ___ that ___ anymore!

(2.3) I ___ said, ___ "The mail ___ just came."

If you heard or read sentence (2.1) without being able to look ahead, you would probably have a single interpretation at each stage of input, something like, *A large mammal with black fur // A large mammal with black fur is ingesting // A large mammal with black fur is ingesting an aquatic animal*. You would probably not consider the possibility of nonliteral word senses, implicatures, sarcasm, humor, and the like. There is no need to invoke such extra reasoning since your basic knowledge of language and the world led to an interpretation that worked just fine.

By contrast, (2.2) requires more effort, and more context, to interpret. Since the small simians we call monkeys cannot make promises, either *monkey* or *promise* must be nonliteral. *Monkey* is a more obvious choice since people are often referred to by the name of an animal featuring a contextually relevant characteristic. Our sentence could, for example, be said jokingly about a child who is swinging dangerously from a jungle gym. In addition, although a basic interpretation can be gleaned from the sentence without contextual grounding, its full interpretation requires determining the referents for *my*, *he*, and *that*.

Example (2.3), when taken out of context, has two instances of residual ambiguity: the identity of *I* and the force of *I said*. The latter can be used

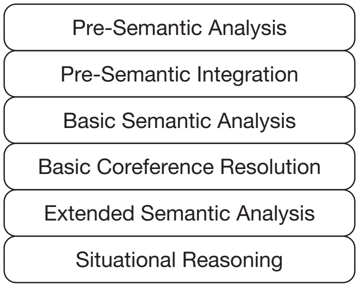

Figure 2.2
Stages of vertical context available during NLU by LEIAs.

- when the interlocutor fails to hear the original utterance, as in a noisy room or over a bad phone connection;
- as part of a story: "I said, 'The mail just came.' And he suddenly leaps out of his chair and barrels out the door!"; and
- to emphasize an indirect speech act that was not acted on originally: "The mail just came. [No reaction] I *said*, the mail just came." (The implication is that the interlocutor is supposed to go fetch it.)

All these interpretations can be computed from general lexical, semantic, and pragmatic knowledge, but choosing among them requires additional features from the speech context.

The point of these examples is that it would make no more sense to have LEIAs invoke deep reasoning to analyze (2.1) than it would to expect them to understand the pragmatic force of *I said* in (2.3) without access to contextual features. Accordingly, a key aspect of the intelligence of intelligent agents is their ability to independently determine which resources to leverage, when, and why, as well as what constitutes a sufficient analysis.

We have found it useful to organize vertical context into the six processing stages shown in figure 2.2.

These stages are detailed in chapters 3–7 but we sketch them below to serve the introductory goals of this chapter.

1. Pre-Semantic Analysis covers preprocessing and syntactic parsing. It is carried out by an externally developed tool set and includes part-of-speech tagging, morphological analysis, constituent and dependency parsing, and named-entity recognition.
2. Pre-Semantic Integration adapts the abovementioned heuristic evidence so that it can best serve semantic/pragmatic analysis. Component functions establish linkings between input strings and lexical senses; reambiguate certain decisions that are inherently semantic and, therefore, cannot be made confidently during syntactic analysis (prepositional phrase attachments, nominal-compound bracketing, and preposition/

particle tagging); attempt to recover from noncanonical syntactic parses; and carry out the first stage of new-word learning.

3. Basic Semantic Analysis carries out lexical disambiguation and semantic dependency determination for sentences taken individually. It computes what some call "sentence semantics" by using the static knowledge recorded in the lexicon and ontology. It often results in residual ambiguity (multiple candidate analyses) and often contains underspecifications (e.g., *he* is some male animal whose identity must be contextually determined).
4. Basic Coreference Resolution includes a large number of functions aimed at identifying textual coreferents for overt and elided referring expressions. During this stage, the agent also reconsiders lexical disambiguation decisions in conjunction with what it has learned about coreference relations.
5. Extended Semantic Analysis invokes more knowledge bases and more reasoners to improve the semantic/pragmatic analysis of not only individual sentences but also multisentence discourses.
6. Situational Reasoning attempts to compute all outstanding aspects of contextual meaning using the agent's situational awareness—that is, its interpretation of nonlinguistic percepts, its knowledge about its own and its interlocutor's plans and goals, its mindreading of the interlocutor, and more.

For purposes of this introduction, two aspects of these stages deserve further comment: (a) the first five stages form a functional unit, for both theoretical and practical reasons, and (b) the conclusion of each stage represents a decision point for the agent. We consider these issues in turn.

a. *The first five stages form a functional unit* since they represent all of the language analysis that the agent can bring to bear without having specialized knowledge of the domain or carrying out situational reasoning, which can go far beyond language per se (involving plans, goals, mindreading, and so on). We find Harris's (in press) notion of "semantic value" a useful description of this level of processing—although, his definition assumes that sentences are considered individually, whereas LEIAs can analyze multisentence inputs together using these first five stages. Harris defines a sentence's semantic value as "not its content but a partial and defeasible constraint on what it can be used to say."

For example, *Give him a shot* can refer to a medical injection, a scoring opportunity, the imbibing of a portion of alcohol, or the opportunity to carry out some unnamed action that the person in question has not yet had the opportunity to try. (The fact that, given this input in isolation, individual people might not recognize the ambiguity and might default to a particular analysis reflects aspects of their individual minds, experiences, and interests.) If the agent has access to the preceding context, and if that context happens to mention something from the medical realm (e.g., a patient, a hospital), then the agent will prefer the *injection* interpretation. But if this evidence is not available, then residual ambiguity is entirely correct as the result of the fifth stage of processing.

Apart from being theoretically justified, there is a practical reason to bunch stages 1–5: these, but not stage 6, can be profitably applied to texts in the open domain as a means of validating and enhancing microtheories (see chapter 9).

b. *The conclusion of each stage represents a decision point for the agent*, represented by the control flow in figure 2.3. Decisions about actionability rely on the particular plans and goals of a particular agent at a particular time—a topic discussed in chapter 8.

Although it is, in some respects, premature to describe the agent's post-stage decision-making before detailing what happens in each stage, we have found in teaching this material that it is important to motivate, right from the outset, why the stages are formally delineated rather than merged into one all-encompassing program. We will provide this motivation using examples, with the caveat that readers are not expected to understand all the details. Instead, they should aim to understand the gist—and then plan to return to this section later on, having absorbed the material in chapters 3–7.

Specifically, each subsection below provides multiple examples of the decision points labeled 1–6 in figure 2.4.

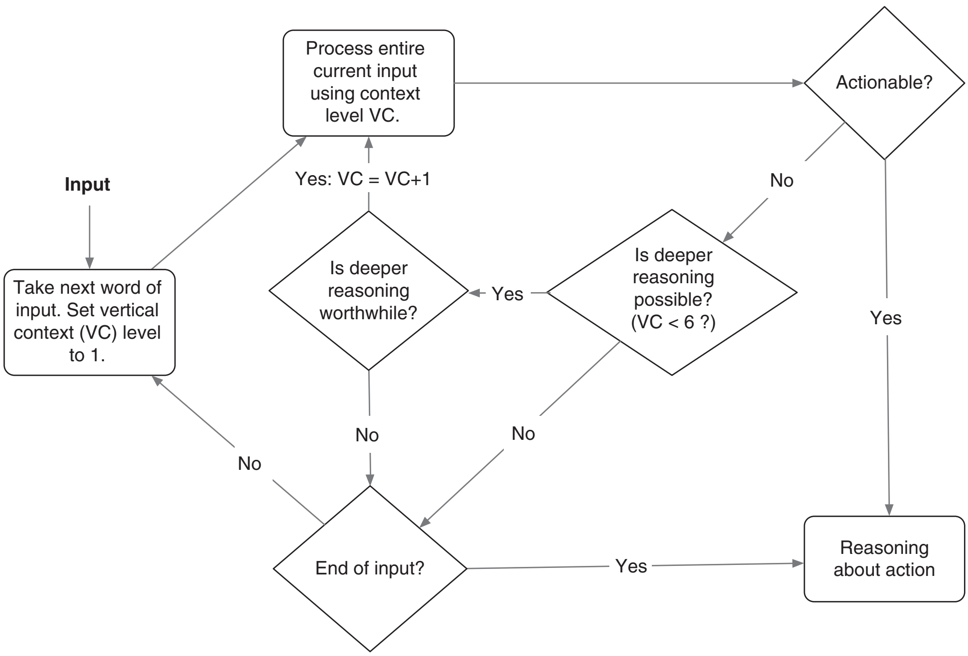

Figure 2.3
The control flow of decision-making during semantic analysis.

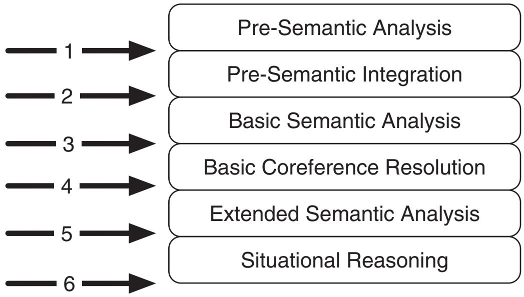

Figure 2.4
Decision points during vertical-incremental processing.

Some of the examples are relevant for most applications, whereas others envision applications of a particular profile. For example, high-risk applications require the agent to be more confident in its analyses than do low-risk applications. Similarly, applications that are expected to involve extensive off-topic exchanges impose different challenges than those in which participants are expected to reliably stay on topic.

2.5.1 Decision-Making after Pre-Semantic Analysis

Example 1. The agent is a furniture-assembly robot. It has multiple human collaborators who chat about this and that to pass the time. The agent is modeled to "skim" what it hears and only semantically analyze inputs that might be task relevant. To operationalize this skimming, the agent has a precompiled list of words and phrases of interest. The list was generated by searching the lexicon for all words and/or phrases that map to concepts of interest. Since the task is not urgent, the agent does not use horizontal incrementality—it waits until the end of each sentence to do any processing at all. At that point, it runs Pre-Semantic Analysis and, thanks to the morphological analyzer, it obtains base (dictionary) forms for all input words. It compares those with its list of words and phrases of interest. If there is little to no overlap, it treats the input as actionable with the action being *Ignore this utterance; it is outside of purview*. This will be the agent's decision when, for example, it hears, "Did you see the Pens last night? The third period was a heartbreaker!"[13]
Example 2. The agent is a military robot engaged in combat. It works with only one human teammate at a time, so there is no chance of off-topic conversation. The task is maximally high-risk and time-sensitive. The agent must understand absolutely everything its human teammate says. The agent analyzes each word immediately (horizontal incrementality) and as deeply as possible (vertical incrementality). At the stage of Pre-Semantic Analysis, if any word of input cannot be clearly recognized (e.g., due to background noise), the agent interrupts immediately for clarification.
Example 3. The agent is tasked with expanding its lexicon off-line. In order to focus on specific words, it must skim a lot of text. The morphological analyzer and part-of-speech

tagger, which count among the preprocessing tools, can identify sentences that contain word forms of interest. Having compiled a set of potentially interesting contexts using these shallow analysis methods, the agent can then apply deeper analysis to them in service of learning.

2.5.2 Decision-Making after Pre-Semantic Integration

Example 1. Returning to our furniture-building robot, we said that one way to implement skimming was to precompile a list of words and phrases of interest and check new inputs against it. Another way to do this (without precompiling a list) is to carry out Pre-Semantic Integration, during which the agent looks up the words of input in its lexicon and can determine if they map to concepts of interest.

Example 2. Returning to our military robot who must fully understand everything, if it realizes, as a result of lexical lookup, that a word is missing from its lexicon, it can interrupt its human partner and immediately initiate a new-word-learning dialog rather than engage in the normal, multistage process of new-word learning. The latter process would require that the agent wait until the end of the utterance to attempt the learning—which might be too long, depending on the urgency of the application.

Example 3. Three of the procedures launched during Pre-Semantic Integration attempt to recover from bad (irregular, incomplete) parses. If those procedures (see sections 3.2.2, 3.2.4, and 3.2.6) work reasonably well, then processing can proceed as usual. However, if they don't, then the typical syntax-informs-semantics approach to NLU will not work. In such cases, the agent will skip stages 3–5 and proceed directly to stage 6, Situational Reasoning, where it will attempt to cobble together the intended meaning using minimal syntactic evidence (essentially, noun-phrase boundaries and word order) supported by extensive semantic and pragmatic analysis. This decision is used by all agents in all applications since there is no way to strong-arm stages 3–5 given an irreparably bad syntactic parse.

Example 4. Many agents will be tasked with learning new words on the fly. If there is a maximum of one unknown word per clause, this learning process has a chance of working well. If, by contrast, there are multiple unknown words in a given clause, it is unlikely that new-word learning will produce useful results. So, upon detecting the multiple-unknown-words case, a learning agent can choose to either ignore the sentence (under the assumption that so many unknown words would make it out of purview) or tell its human teammate that the input contains too many unknown words to be treated successfully.

2.5.3 Decision-Making after Basic Semantic Analysis

Example 1. Back to our chair-building robot, one of its tasks is to determine which inputs can be ignored because they are out of purview. We already saw two methods of making this determination. This stage presents us with a third, which offers even higher confidence. The agent can go ahead and carry out Basic Semantic Analysis—which usually results in multiple candidate analyses—and check whether any of them aligns with its in-purview ontological scripts. This method is more reliable than previous ones because it looks not only for concepts of interest but for combinations of concepts of interest. For example, the humans in the scene might be having off-topic conversations about chairs—for example, they might be discussing the selection of chairs for someone's new home. So the mention of chairs does not guarantee that the input is relevant to the agent.

Example 2. For some inputs, Basic Semantic Analysis is sufficient to generate a full and confident interpretation—as for "Chairs are so easy to build!" This sentence matches a construction (X is (ADV) easy to Y) that guides semantic analysis. The words *chair* and *build* are ambiguous, but the highest-scoring semantic analysis combines the physical meaning of *build* with the furniture meaning of *chair*. None of the entities requires coreference, and there are no indicators that a nonliteral interpretation should be sought, so the analysis is finished and the agent will delve no deeper.

Example 3. Consider, again, the lexicon-expansion agent that has identified sentences of interest (thanks to decision-making after Pre-Semantic Analysis) and must now analyze them more deeply. In most cases, Basic Semantic Analysis is sufficient to support new-word learning. The reason is that, given a large corpus, the agent can choose to avoid difficult examples, such as those that require coreference resolution. For example, it can learn the meaning of *cupuacu* from the simple and direct "Some people think that *cupuacu* tastes good" rather than attempt to figure it out from the trickier "There are many exotic fruits that most Americans have never tried. Take, for example, *cupuacu*."

2.5.4 Decision-Making after Basic Coreference Resolution

Example 1. Basic Semantic Analysis was sufficient to generate a full and confident interpretation for some inputs (e.g., "Chairs are so easy to build!"). Basic Coreference Resolution expands the inventory of inputs that are fully and confidently interpreted. For example, if we modify our example to "Chairs are great; they are so easy to build!", establishing the coreference between *chairs* and *they* is all that is needed on top of Basic Semantic Analysis to generate a full and confident interpretation. Similarly, verb phrase ellipsis in the following example can be resolved during Basic Coreference Resolution, resulting in a full interpretation of "<u>Disassembling chairs</u> is frustrating but <u>it</u>'s sometimes necessary."

Example 2. We said earlier that our lexicon-expansion agent will generally avoid inputs that contain underspecified referring expressions, such as pronouns. However, it can choose to include such inputs in its corpus for learning if (a) the coreference relations can be easily and confidently established or (b) the word is so rare that the only available examples include pronouns. Returning to our example with *cupuacu*, a sentence like "Some people like <u>*cupuacu*</u> and eat <u>it</u> regularly," the coreference between *it* and *cupuacu* can be reliably established, so that the agent can use the understood proposition *they eat cupuacu* as its example for learning.

2.5.5 Decision-Making after Extended Semantic Analysis

Example 1. Continuing with the lexicon-expansion agent, after Basic Semantic Analysis, even supplemented by Basic Coreference Resolution, the agent will often be dealing with as-yet incomplete analyses involving residual ambiguity (i.e., multiple viable candidates), incongruity (no high-scoring candidates), or underspecification. This stage offers many functions to improve such analyses. For example, the agent has methods to select the correct meaning of *hand* in "If you see a mechanical clock on the wall and its *hands* are not moving, try winding it and resetting it." It will also be able to fully analyze the nominal compound *physician neighbor* found in an example like "My physician neighbor came straight over when I called saying that my dog was bleeding" (the compound means HUMAN (HAS-SOCIAL-ROLE PHYSICIAN, NEIGHBOR)). More accurate analyses support more correct and more specific new-word learning.

Example 2. These first five stages of analysis can be launched on texts in any domain, outside of comprehensive agent applications. This is because a considerable amount of reasoning related to language understanding relies on basic knowledge of lexicon and ontology. Therefore, whenever language understanding is undertaken outside of a full-fledged agent application, if the application is not time-sensitive, the most natural decision is to run the inputs through Extended Semantic Analysis to achieve the best possible outcome.

2.5.6 Decision-Making after Situational Reasoning

Example 1. The chair-building robot will need to ground many referring expressions in the inputs that are relevant to it: for example, "Grab that hammer and give it to me." It will also need to carry out such grounding as it continues the task of determining which inputs are off-topic. For example, someone's dialog turn might be, "I think we should move all the chairs there first, since they are light," which may or may not be relevant to the agent depending on the referents for *we*, *the chairs*, and *there*. Once these referents have been established, the agent can determine that, in the following context, this utterance is outside of purview: "My wife and I decided to remodel the first floor of our house. When we refinish the floors, we are thinking of putting all the furniture in the basement to get it out of the way. I think we should move all the chairs there first, since they are light." This is part of Situational Reasoning, not Basic Coreference Resolution, because of the grounding needs.

Example 2. Our envisioned military robot is likely to process many inputs that require full reference resolution, defined as linking mentioned entities to both the real-world environment and to its short-term memory. These are both part of Situational Reasoning. For example, if the robot is told, "Pick up that grenade and throw it into the building," the robot will need to ground the referents of *that grenade* and *the building* (it will have coreferred *it* with *the grenade* during Basic Coreference Resolution). If the agent has any doubt about its interpretation, it will need to clarify with its human teammate.

Example 3. Human dialog is often syntactically irregular and fragmentary. As mentioned with respect to the decision-making after stage 2, syntactically irregular inputs that cannot be normalized or worked around to a reasonable degree will sidestep stages 3–5. At this stage, the agent will try to make sense of them by cobbling together candidate semantic analyses and comparing those candidates against the expectations of task-relevant scripts. For example, upon hearing the input, "That screwdriver … no, wait, maybe a hammer is better? Yeah, give me one," the agent will need to understand that the intended command is "Give me a hammer." If the agent is capable of doing only a limited number of things, among which is giving a human a hammer, it can proceed to act on this interpretation.

This concludes our brief and, in some sense, premature overview of agent decision-making during language analysis. The goals were modest: to provide an initial motivation for dividing the process of NLU into distinct stages and to highlight the kinds of decisions the agent must make to proceed after each one.

2.6 Microtheories

We use the term *microtheories* to refer to computational linguistic models that address individual issues of language processing, such as word sense disambiguation, coreference resolution, and indirect speech act interpretation—to name just a few. Microtheory development involves an initial top-down analysis of the whole problem space (or a realistic approximation of it) followed by deeper, algorithm-supported treatments that are implemented and iteratively enhanced. Among the methodological preferences in building microtheories are striving for a natural, waste-free progression over time (rather than hacking partial solutions that will necessarily be thrown away) and prioritizing phenomena that are most readily treatable and/or most urgent for a particular application or research goal.

It would be impossible to overstate how great a distance there can be between simple and difficult manifestations of a linguistic phenomenon. Consider nominal compounding. Analyzing a compound can be as simple as looking up a sense stored in the lexicon (*coffee cup*) or as difficult as attempting to learn the meanings of two unknown words on the fly before semantically combining them in a context-appropriate way. The difference in timelines for achieving confident results across so broad a spectrum underscores why it is so important to at least sketch out the entire problem space when initially developing a microtheory.

Saying that linguistic phenomena will be treated over time does not imply that we expect each one to submit, quickly and gracefully. Quite the opposite. The problems will only get more difficult as we approach the hardest 5% of instances. Our objective is to make LEIAs well-rounded enough in their situational understanding to be able to adequately deal with the hardest cases in the same ways that a person does when faced with uninterpretable content (even though what is uninterpretable for LEIAs will, in many cases, be different from what is uninterpretable for people).

Computational linguistics, like its parent discipline, computational cognitive science, is an applied science. This means that it leverages scientific understanding in service of practical applications. However, computational linguistics presents not a clear dichotomy between *theory* and *system*[14] but more of a trichotomy between *theory*, *model*, and *system*. As a first approximation, theories in cognitive science are abstract and formal statements about how human cognition works; models account for real data in computable ways and are influenced as much by practical considerations as by theoretical insights; systems, for their part, implement models within the real-world constraints of existing technologies. Since this trichotomy is at the heart of our story, let us flesh it out just a bit further.[15]

Theories. The most general statement of our view is that theories attempt to explain and reflect reality as it is, albeit with great latitude for underspecification. Another important property of theories for us is that they are not bound by practical concerns such as computability or the attainability of prerequisites. We share the position, formulated in Winther (2016), that "laws of nature are rarely true and epistemically weak. Theory as a collection of laws cannot, therefore, support the many kinds of inferences and explanations that we have come to expect it to license." This position ascends to Cartwright's (1983) view that "to explain a phenomenon is to find a model that fits it into the basic framework of the theory and that thus allows us to derive analogues for the messy and complicated phenomenological laws which are true of it" (p. 152).

Theories guide developers' thinking in developing models and interpreting their nature, output, and expectations. In our work we are guided by the theory of Ontological Semantics (Nirenburg & Raskin, 2004) that proposes the major knowledge and processing components of the phenomena in its purview, which is language understanding. However, the lion's share of our work is on developing models (microtheories) and systems.

Models. Computational cognitive models of language processing formally describe specific linguistic phenomena as well as methods for LEIAs to process occurrences of these phenomena in texts.[16] The most important property of such models is that they must be computable. This means that they must rely exclusively on types of input (e.g., property values) that can actually be computed using technologies available at the time of model construction. If some feature that plays a key role in a theory cannot be computed, then it either must be replaced by a computable proxy, if such exists, or it must be excluded from the model. In other words, models, unlike theories, must include concrete decision algorithms and computable heuristics. To take just one example from the realm of pronominal coreference, although the notions *topic* and *comment* figure prominently in theoretical-linguistic descriptions of coreference, they do not serve the modeling enterprise since their values cannot be reliably computed in the general case. (To date, no adequate model has been proposed for deriving such values.)

Models must account for the widest possible swath of data involving a particular linguistic phenomenon—which is a far cry from the neat and orderly examples found in dictionaries, grammars, and textbooks.

Models should embrace well-selected simplifications, drawing from the collective experience in human-inspired machine reasoning, which has shown that it is counterproductive to populate decision functions with innumerable parameters whose myriad interactions cannot be adequately accounted for (Kahneman, 2011).

Models of natural language should reflect the fact that people are far from perfect both in generating and in understanding utterances, and yet successful communication is the norm rather than the exception. So, models of language processing need to account for both the widespread imperfection and the overwhelming success of language use. The notions of *cognitive load* and *actionability* are instrumental in capturing this aspect of modeling. Cognitive load describes how much effort humans have to expend to carry out a mental task. As a first approximation, a low cognitive load for people should translate into a simpler processing challenge for machines and, accordingly, a higher confidence in the outcome. Of course, this is an idealization since certain analysis tasks that are simple for people (such as reasoning by analogy) are quite difficult for machines; however, the basic insight remains valid. Actionability captures the idea that people can often get by with an imperfect and incomplete understanding of both language and situations. So, the approximations of cognitive load, and the associated confidence metrics, do not carry an absolute interpretation. For example, in the context of off-task chitchat, an agent might decide to simply keep listening if it doesn't understand exactly what its human partner is saying since the risk of incomplete understanding is little to none. By contrast, in the context of military combat, anything less than full confidence in the interpretation of an order to aggress will necessarily lead to a clarification subdialog to avoid a potentially catastrophic error.

Finally, models must operationalize the factors identified as most important by the theory. Cognitive load and actionability provide useful illustrations of this requirement. The cognitive load of interpreting a given input can be estimated using a function that considers the number and complexity of each contributing language-analysis task. Let us consider one example from each end of the complexity spectrum. The sentence *John ate an apple* will result in a low-complexity, high-confidence analysis if the given language understanding system generates a single, canonical syntactic parse, finds only one sense of *John* and one sense of *apple* in its lexicon, and can readily disambiguate between multiple senses of *eat* given the fact that only one of them aligns with a human agent and an ingestible theme. At the other end of the complexity (and confidence) spectrum is the analysis of a long sentence that contains multiple unknown words, does not yield a canonical syntactic parse, and offers multiple similarly scoring semantic analyses of separate chunks of input.

The above demonstrates the importance of gauging the simplicity of language material. Whether we are building a model of verb phrase ellipsis resolution, nominal compound interpretation, lexical disambiguation, or new-word learning, we can start by asking, Which kinds of attested occurrences of these phenomena are *simple*, and which feature values manifest that simplicity? Then we can start our model development with the simpler phenomena and proceed to the more complicated ones once the basics of the nascent microtheory have been sketched.

"Simpler-first" modeling can be guided by various linguistic principles, such as parallelism and prefabrication (e.g., remembered expressions and constructions). Consider, in this regard, the pair of examples (2.4) and (2.5), which illustrate the type of verbal ellipsis called *gapping*.

(2.4) Delilah is studying Spanish and Dana __, French.

(2.5) ? Delilah is studying Spanish and my car mechanic, who I've been going to for years, __, fuel-injection systems.

Gapping is best treated as a *construction* (a prefabricated unit) that requires the overt elements in each conjunct (i.e., the arguments and adjuncts) to be syntactically and semantically *parallel*. It also requires that the sentence be relatively *simple*. The infelicity of (2.5), indicated by the question mark, results from the lack of simplicity, the lack of syntactic parallelism (the second clause includes a relative clause not present in the first), and the lack of semantic parallelism (languages and fuel-injection systems are hardly comparable).

Simpler-first modeling carries one absolute requirement: that the models enable agents to independently determine which examples are covered by the model and with what confidence value. There is no oracle to tell agents that they *can* understand *this* example, but *that* example is too hard. Every model described in this book includes methods for automatically detecting which examples are covered and with what confidence. (This topic is explored in particular depth in section 9.3.) Of course, agents need to treat every input, by hook or by crook, but the treatment can involve "consciously" generating low-confidence analyses. How agents will act on such analyses will be decided in non-NLU modules of the cognitive architecture.

Computing confidence in overall language analysis is not a simple matter. It requires establishing the relative importance of all subtasks that contribute to overall processing and accounting for their interactions. For example, even if an input's syntactic parse is suboptimal, an agent can be confident in a candidate interpretation if the latter works out semantically and is situationally appropriate—meaning that it aligns with the agent's expectations about what should happen next in a workflow script.

As concerns actionability, it can only be judged on the basis of an agent's assessment of its current plans and goals, its assessment of the risk of a mistake, and so on—which means that the modeling of language must necessarily be integrated with the modeling of all other cognitive capabilities (see chapters 7 and 8).

Systems. The transition from models to systems moves us yet another step away from the neat and abstract world of theory. Models are dedicated to particular phenomena, while the overall task of natural language understanding involves the treatment of many phenomena in a single process. Thus, the first challenge of building comprehensive NLU systems is integrating the computational realizations of the models of individual phenomena. This, in turn, requires managing inevitable cross-model incompatibilities. Since the idea of cross-model incompatibilities might not be self-evident, we will unpack it.

Any program of R&D must take into account economy of effort. In the realm of knowledge-based NLU, if components and tools for computing certain heuristic values exist, then developers should at least consider using them. However, importation comes at a cost. Externally developed components and tools are likely to implement different explicit or implicit linguistic models, thus requiring an added integration effort. For example, different systems rely on different inventories of parts of speech, syntactic constituents, and semantic dependencies. So, if an off-the-shelf preprocessor and syntactic parser are to be imported into an NLU system, the form and substance of the primitives in the source and target models must be aligned—which not only requires a significant effort but also often forces modifications to the target model, not necessarily improving it.

There is no generalized solution to the problem of cross-model incompatibility since there is no single correct answer to many problems of language analysis—and humans are quite naturally predisposed to hold fast to their individual preferences. So, dynamic model alignment is an imperative of developing computational-linguistic systems that must be proactively managed. However, its cost cannot be underestimated. We are talking here not only about the initial integration effort. The output of imported modules often requires modification, which requires additional processing. (See section 3.2.5 for examples.) In fact, the cost of importing processors has strongly influenced our decision to develop most of our models and systems in-house.

Another challenge of system building is that all language processing subsystems—be they imported or developed in-house—are error-prone. Even the simplest of capabilities, such as part-of-speech tagging, are far from perfect at the current state of the art. This means that downstream components must anticipate the possibility of upstream errors and prepare to manage the overall cascading of errors—all of which represents a conceptual distancing from the model that is being implemented.

Because of the abovementioned and other practical considerations, implemented systems are unlikely to precisely mirror the models they implement. This complicates the task of assessing the quality of models. If one were to seek a "pure" evaluation of a model, the model would have to be tested under the unrealistic precondition that all upstream results used by the system that implemented the model were correct. In that case, any errors would be confidently attributed to the model itself. However, meeting this precondition typically requires human intervention, and introducing such intervention into the process would render the system not truly computational in any interesting or useful sense of the word. The system would amount to a theory-model hybrid rather than a computational linguistic system. So, as long as one insists that systems be fully automatic (we do), any evaluation will be namely an evaluation of a system, and, in the best case, it will provide useful insights into the quality of the underlying model.

To summarize this section on microtheories: Our microtheories are explanatory, broad-coverage, heuristic-supported treatments of language phenomena that are intended to be implemented and enhanced over time.

An obvious question is, *Why not import microtheories?* It would be a boon if we could, but, unfortunately, the majority of published linguistic descriptions either are not precise enough to be implemented or rely on preconditions that cannot be automatically fulfilled—as discussed in chapter 1. If our vision of Linguistics for the Age of AI takes hold, linguists *will* take up the challenge of developing these kinds of microtheories, which we will be only too happy to import.

2.7 "Golden" Text Meaning Representations

The idea of "golden" (also known as gold and gold-standard) TMRs comes from the domain of corpus annotation. Golden annotations are those whose correctness has been attested by people, either because people manually created the annotations to begin with or because they checked and corrected the results of automatic annotation. Golden annotations are a foundation of supervised machine learning.

Analogously, we call a TMR golden if, in generating it, (a) the LEIA has leveraged the current state of the knowledge bases correctly and (b) those knowledge bases, along with supporting NLU rule sets, were sufficient to generate a high-quality analysis (McShane et al., 2005c).

However, it is not the case that the LEIA will always, over time, generate the same golden TMR for a given input. The reason is that knowledge engineering is an ongoing process, and knowledge can be recorded in different ways and at different grain sizes. This means that the golden TMR generated for a given input in 2020 might not be exactly identical to the one generated in 2025.

For example, at the time of writing, LEIAs have not been used in applications specifically involving hats, so the English words for all kinds of hats (*beret*, *fedora*, *baseball cap*, and so on) are mapped to the concept HAT with no distinguishing property values listed. After all, it takes time and energy to record all those property values. The current analyzer, therefore, generates the same TMR for the sentence variants "The hat <beret, fedora, baseball cap> [i.e., HAT] blew off his head." However, if a hat manufacturer came along in 2024 with a request for a hat-sales app, then the ontological tree for HAT might be expanded into all subtypes with their relevant property values. The 2025 analyzer would then generate different analyses for the variants of our hat example.

All of this might not seem important until one considers the potential utility of TMRs for the long-term support of both LEIAs themselves and the NLP community at large. As we describe in section 6.1.6, LEIAs can use their repository of past TMRs to help with certain aspects of analyzing new inputs. However, the sliding scale of correctness must factor into this analogical reasoning. Similarly, a sufficiently large repository of TMRs, created over time, could seed machine learning in service of subsequent LEIA-style NLU, in the way that manual annotation efforts currently do—but again, with the caveat that levels of precision might be different. In short, golden TMRs could serve as excellent, semantically oriented corpus annotations, but they would have to be linked to the specific version of each knowledge base used to generate them, since *correct* with one knowledge base might not be with another.

An interesting issue with respect to golden TMRs involves paraphrase, both in language and in the ontological metalanguage. We address this topic in the deep dive in section 2.8.4.

2.8 Deep Dives

2.8.1 The LEIA Knowledge Representation Language versus Other Options

Over the history of AI, many opinions about the optimal relationship between natural language (NL) and knowledge representation languages (KRLs) have been expressed, and many approaches to KRLs have been tried. There are at least three substantially different ones, along with many variations on the themes. We briefly review them below.

1. *NL →KRL →NL translation*. This is the approach advocated in this book and, for purposes of this survey, it is sufficient to list its key advantages and challenges. *Advantages*: The NL input and output are readable by people. The KRL is readable by trained people. NL knowledge bases are available as the source of translations into the KRL. The KRL is suitable for reasoners. The KRL representations are language independent, thus fostering multilingual applications. *Challenges*: The KRL must be expressive and influenced by NL. The translation requires large, high-quality knowledge bases, high-quality analyzers and generators, and extensive reasoning about language, the world, and the situation.

2. *NL is KRL (or KRL is NL)*. This position ascends to Wittgenstein (1953) and essentially declares that the symbols used in KRLs are ultimately taken from NLs and cannot be taken from anywhere else. As a result, KRLs retain NL-like features, such as ambiguity and vagueness, no matter how carefully these languages are designed. The NL-is-KRL movement among computational logicians and proponents of controlled languages is relatively recent and has been formulated as a research program: How to make KRLs more NL-like? Yorick Wilks believes that KRLs are already, in the final analysis, NLs. He has consistently argued (e.g., Wilks 1975, 2009) that knowledge representations are, by nature, ambiguous and vague and that it is, in principle, impossible to eliminate such “language-like features” as ambiguity from ontologies and conceptual structures. This position sounds the death knell for standard automatic reasoning techniques because it essentially states that any automatic reasoning will be indeterminate.

At the same time, Wilks claims, and rightly so, that ambiguous, incomplete, and inconsistent knowledge resources can be, and still are, useful for NLP. As his main example, he cites WordNet (Miller, 1995), as it has been used widely by corpus-based NLP practitioners even though it is demonstrably challenging when used as a knowledge base for NLP (Nirenburg et al., 2004). The catch is that the types of applications Wilks has in mind rely only partially on semantics. Take the example of a personal conversational assistant (Wilks, 2004; Wilks et al., 2011), which can, to a degree, fake understanding and, when needed, change the topic of conversation to something that it can better handle. This is an admirable sleight-of-hand strategy that works in applications whose purpose is largely phatic communication. Strategically similar approaches can pass the responsibility for understanding to a human in the loop: for example, people can often make sense of noisy output from machine translation and summarization systems. However, such detour strategies will not work when understanding by the intelligent agent is crucial. We have commented in detail on the NL-is-KRL opinion in Nirenburg and Wilks (2001) and Nirenburg (2010); here we will limit our remarks to a few relevant methodological points.

While Wilks cited the extraction and manipulation of text meaning as the major scientific objective of his work, ambiguity of representation was not a central issue for him. The desire to nudge the evaluation results of systems like word sense disambiguation engines into the 90% range (cf. Ide & Wilks, 2006) led him to claim that, for NLP, only word sense distinctions at the coarse-grained level of homographs are important.[17] Such a claim may work in the world of Semeval (https://www.wikiwand.com/en/SemEval) and similar competitions but, in reality, the situation with sense delimitation is much murkier.

For example, the English word *operation* has eleven senses in the *American Heritage Dictionary* and (so far) three senses in the LEIA's lexicon—roughly, *military operation*, *surgery*, and *general state of functioning*. In the LEIA's ontological metalanguage, different concepts correspond to these meanings, each with its own set of properties and value sets. If, by contrast, this three-way ambiguity were retained in the representation, then, to gain more information about *the operation*, the reasoner would not know whether to ask, "Was general anesthesia administered?" or "Was a general in command?" Of course, it is entirely possible that, at any given time, some of the property-based distinctions needed to avoid confusing the reasoning engine have not yet been introduced. It follows that, if certain distinctions are not required for any reasoning purposes, such benign ambiguities may be retained in the representation. This is clearly an operational, application-oriented approach, but we have to live with it because the field has not yet come up with a universal theoretical criterion for sense delimitation.

It is reasonable to hope that the balance between short-term and long-term research in NLP and reasoning is on the road to being restored. Even in the currently dominant empiricist research paradigm, researchers recognize that the core prerequisite for the improvement of their application systems (which today achieve only modest results) is not developing better machine learning algorithms that operate on larger sets of training data but, rather, enhancing the types of knowledge used in the processing. The terminology they prefer is judicious selection of distinguishing features on which to base the comparisons and classifications of texts. As Manning (2004) notes, "In the context of language, doing 'feature engineering' is otherwise known as doing linguistics. A distinctive aspect of language processing problems is that the space of interesting and useful features that one could extract is usually effectively unbounded. All one needs is enough linguistic insight and time to build those features (and enough data to estimate them effectively)." These features are, in practice, the major building blocks of the metalanguage of representation of text meaning and, therefore, of KRL.

To summarize, the advantages of *NL is KRL* are that people find it easy to use and NL knowledge bases are available. The main problem is that reasoning directly in NL amounts to either matching uninterpreted strings or operating in an artificially constrained space of "allowed" lexical senses and syntactic constructions (see next section). This is why the machine reasoning community has for decades been writing inputs to its systems by hand.

3. *Controlled NL as KRL*. The notion of controlled languages ascends at least to the "basic English" of Ogden (1934). Controlled languages have a restricted lexicon, syntax, and semantics. Dozens of controlled languages, deriving from many natural languages, have been developed over the years to serve computer applications. A typical application involves using a controlled language to write a document (e.g., a product manual) in order to facilitate its automatic translation into other languages. Presumably, the controlled-language text will contain fewer lexical, grammatical, semantic, and other ambiguities capable of causing translation errors. Controlled languages are usually discussed together with authoring tools of various kinds—spell checkers, grammar checkers, terminology checkers, style checkers—that alleviate the difficulties that authors face when trying to write in the controlled language.

Controlled languages can also be used as programming languages. According to Sowa (2004), the programming language COBOL is a controlled English. The controlled languages especially relevant to our discussion are computer processable controlled languages (Sukkarieh, 2003), modeled after Pulman's (1996) Computer Processable English. The defining constraint of a computer processable controlled language is "to be capable of being completely syntactically and semantically analysed by a language processing system" (Pulman, 1996). Work in this area involves building tools to facilitate two aspects of the process: authoring texts in the controlled language and carrying out specialized types of knowledge acquisition—for example, compiling NL-NL_c dictionaries that specify which senses of NL words are included in the controlled language (NL_c). One benefit of this approach is that such dictionaries can, if desired, be completely user dependent, which means that different kinds of reasoning will be supported by the same general apparatus using these idiolects of NL_c.

A large number of computer processable controlled languages have been proposed, among them the language used in the KANT/KANTOO MT project (Nyberg & Mitamura, 1996), Boeing's Computer Processable Language (P. Clark et al., 2009), PENG Light Processable ENGlish (http://web.science.mq.edu.au/~rolfs/PENG-Light.html), the Controlled English to

Logic Translation (CELT) language (Pease & Murray, 2003), Common Logic Controlled English (Sowa, 2004), and Attempto (Fuchs et al., 2006). While there are differences between them, strategically all of them conform to the methodology of relying on people, not machines, to disambiguate text. The disambiguated text can be represented in a variant of first-order logic—possibly with some extensions—and used as input to reasoning engines. Having no ontological commitment broadens the opportunity for user-defined applications that can bypass the automatic analysis of open text. However, the research and development devoted to the use of controlled languages is, in our opinion, primarily technology-oriented and contributes little to the long-term goal of creating truly automatic intelligent agents, which is predicated on the capability of understanding unconstrained language.

A variation on the theme is a multistep translation method favored by some logicians. The goal is to constrain NL inputs to reasoners to whatever can be automatically translated into first-order logic. So, the process involves human translation from NL to a controlled NL, for which a parser into KRL is available. As such, having a controlled NL (NL_c) and its associated parser becomes equivalent to having a KRL (e.g., Fuchs et al., 2006; Pease & Murray, 2003; Sukkarieh, 2003; McAllester & Givan, 1992; Pulman, 1996). It is assumed that the overall system output is formulated in NL_c and that this will pose no problems for people. The problem, of course, is that a human must remain in the loop indefinitely.

In sum, the advantage of *controlled NL as KRL* is that texts in a controlled NL can be automatically translated, without loss, into the KRL—and this can be practical in some applications. The problem is that orienting around a controlled language is impractical for the general case of deriving knowledge from text, since the vast majority of texts will never be written in the given controlled language. In addition, the quality of the translation depends on the ambiguity of the controlled NL text, since writing with a complete absence of ambiguity is not always achievable. Finally, writing texts in controlled languages is notoriously difficult for people, even with training, and people must always remain in the loop, being responsible for the very first step: NL-to-controlled-NL translation.

Acknowledging the partial utility of the above options, we believe that fully automatic NL → KRL → NL translation is indispensable for sophisticated, reasoning-oriented applications. Moreover, it is the most scientifically compelling way of approaching the correlation between language and its meaning.

2.8.2 Issues of Ontology

Many issues involving the content and form of ontology are worth discussing, as evidenced by the extensive philosophical and NLP-oriented literature on ontology (see Nirenburg & Raskin, 2004, Chapter 5, for broad discussion; Kendall & McGuinness, 2019, for an engineering-oriented contribution). We constrain the current section to questions that regularly arise with respect to the ontology used by LEIAs.

Issue 1. *How do concepts differ from words?* Ontological concepts, unlike words of a language, are unambiguous: they have exactly one meaning, defined as the combination of their property-facet-value descriptions. For example, the concept SHIP in the LEIA's ontology refers only and exactly to a large sailing vessel, not a spaceship, or the act of transporting something, or any other meaning that can be conveyed by the English word *ship*. Contrast this lack of ambiguity in the LEIA's ontology with the extensive ambiguity found in machine-readable dictionaries (human-oriented dictionaries that are computer accessible) and wordnets (hierarchical inventories of words, e.g., WordNet; Miller, 1995). In all of these resources, a given string can refer to different parts of speech and/or meanings. The problems with using multiply ambiguous lexical resources as a substrate for intelligent agent reasoning are well-documented (see, among many, Bar Hillel, 1970; Ide & Véronis, 1993; Nirenburg et al., 2004).

Issue 2. *How do concepts, stored in the ontology, differ from concept instances, stored in the LEIA's episodic memory?* Concepts represent *types* of objects and events, whereas concept instances represent real-world (or imaginary-world) *examples* of them. In most cases, the distinction between concepts and instances is clear: CAT is a concept, whereas our fourteen-year old multicolor domestic shorthair named Pumpkin is an instance—perhaps recorded as CAT-8 in some particular agent's memory. For LEIAs, this distinction is formally enforced by storing information about concepts in the ontology and information about instances in the episodic memory. Note that some ontologies do not make this distinction: for example, Cyc (Panton et al., 2006) contains at least some instances in its ontology. Although the concept versus instance distribution is usually clear, there are difficult cases. For example, are specific religions, or specific makes of cars, or specific sports teams (whose players and coaches can change over time) concepts or instances? There are arguments in favor of each analysis, and we are making the associated decisions on a case-by-case basis, to the degree needed for our practical work.

Issue 3. *How was the ontology acquired, and how is it improved?* Most of the current ontology was acquired manually in the 1990s. We are not an ontology development shop and, with the exception of project-specific additions—such as extensive scripts for the Maryland Virtual Patient application—we do not pursue general ontology acquisition (though we could mount a large-scale acquisition project if support for that became available). Instead, we focus resources on advancing the science of cognitive modeling and NLU. That being said, we are working toward enabling LEIAs to learn ontological knowledge during their operation (see chapter 8). We agree with the developers of Cyc that "an intelligent system (like a person) learns at the fringes of what it already knows. It follows, therefore, that the more a system knows, the more (and more easily) it can learn new facts related to its existing knowledge" (Panton et al., 2006).

Issue 4. *Why is there no concept for "poodle" in the current ontology?* Since we do not pursue general ontology acquisition, not all concepts that would ideally be described in the LEIA's world model have yet been acquired. In some such cases, the associated English words are treated by listing them as hyponyms of other words in the lexicon. For example, most kinds of dogs are listed as hyponyms of the word *dog* (which is mapped to the concept DOG) in the lexicon, which means that LEIAs will understand that *poodle* and *Basset hound* belong to the class DOG, but they will not know what distinguishes a poodle from a Bassett hound. In other cases, given entities—particularly those belonging to specialized domains, such as airplane mechanics—are entirely missing from our resources, which means that LEIAs will need to treat them as unknown words.

A methodological tenet is that concepts should not be acquired if they do not differ from their parents or siblings with respect to substantive property values—that is, property values apart from those indicating the place of the concept in the ontological hierarchy (IS-A and SUBCLASSES). For example, the concept POODLE should not be acquired unless the acquirer has the time to list at least some of the feature values that differentiate it from other breeds of dog, such those shown below.

POODLE

COLOR	default	black, white, brown, tan
	sem	silver, orange
INTELLIGENCE	default	.9[18]
FRIENDLINESS	default	.9
FUR-TYPE	value	curly-hair
EASE-OF-TRAINING	default	.9

The reason for this tenet is that a concept means only and exactly what its set of property-facet-value triples says it means. As such, if two concepts have the same inventory of property values, then they are, for purposes of agent reasoning, identical.

Issue 5. *Should something as specific as "eating hot liquids with a spoon" be a concept?* If this action needs to be described in great detail, as to support the simulation of a character who eats hot liquids with a spoon, then this has to be a concept whose description will be recorded using a script. In contrast to eating with a fork, eating with a spoon does not permit lifting the object by poking it; and in contrast to eating cold liquids, eating hot liquids often involves cooling the liquid, either by holding a spoonful in the air for a while or by blowing on it. Of course, an EATING-HOT-LIQUIDS-WITH-A-SPOON concept/script could inherit much from its ancestors, but certain things would be locally specified, thus justifying its existence as a separate concept.

Issue 6. *Are all values of all properties locally defined in all frames?* No, for two reasons—one theoretical and the other practical. The theoretical reason is that, in many cases, a particular property cannot be constrained any more narrowly for a child than for its parent. For example, the concept NEUROSURGERY inherits the property-facet-value triple "LOCATION default OPERATING-ROOM" from its parent, SURGERY, because neurosurgery typically happens in the same place as any other surgery. ORAL-SURGERY, by contrast, overrides that default value for LOCATION because it is typically carried out in a DENTIST-OFFICE. On the practical level, we have not had the resources to fully specify every local property value of every concept. In part, this can be automated, but it would also benefit from an industrial-strength manual or semi-automatic acquisition effort.

Issue 7. *Is the LEIA's ontology divided into upper and lower portions?* When developers divide ontologies into upper (top-level, domain-independent) and lower (domain-specific) portions, the goal is to have a single, widely agreed on upper ontology that can serve as the core to which domain-specific ontologies can link.[19] We have not entered into the arena of ontology merging and have made no formal division between upper and lower portions of the LEIA's ontology. In fact, we would argue that the large expenditure of resources on ontology merging is misplaced, since most efforts in that direction do not adequately address core semantic issues that currently cannot be resolved by automatic methods.

Issue 8. *Are other ontologies used when building the LEIA's ontology?* Over the years we have tried to use external resources when building both the lexicon and the ontology. In most cases, we have found that the overhead—learning about the resource, converting formats, and, especially, carrying out quality control—was not justified by the gains.[20] Our use of external resources resonates with the findings of P. Cohen et al. (1999), who attempted to determine experimentally how and to what extent the use of existing ontologies can foster the development of other ontologies. They found that (a) the most help was provided when acquirers were building domain-specific ontologies and (b) many questions still remain about how best to use existing resources for the building of new resources.

Issue 9. *Why is the LEIA's ontology not available as freeware?* As mentioned earlier, we use the ontology as a substrate for research; we are not developing it as a resource. It has idiosyncrasies that are not documented, and, most importantly, we are not in a position to provide user support.

Issue 10. *Doesn't every culture, and even every individual person, have a different ontology?* Perhaps, but the importance of this consideration for LEIA development is quite limited and specific. That is, we are not especially interested in how many colors are recognized as basic by speakers of different languages or whether the nuances of different verbs of motion should be recorded as different concepts or be described in the lexical senses of the associated words. In this, we agree with McWhorter's (2016) critique of Neo-Whorfianism to the effect that minor crosslinguistic differences should not be blown out of proportion. What *is* important for LEIAs is that they can learn new knowledge, and even learn incorrect knowledge (like people do) during their operation—all of which can have interesting consequences.

Issue 11. *Are there other examples of ontologically grounded deep NLU systems?* Yes, but they better serve as points of juxtaposition than comparison. For example, Cimiano, Unger, and McCrae (2014; hereafter, CU&M) describe an approach to NLU with a strong reliance on handcrafted ontological and lexical knowledge bases. They use a domain-specific ontological model to drive lexical acquisition, and they constrain the interpretations of inputs to those applicable to the domain.

> The interpretation process generates only interpretations that are compatible with the ontology (in terms of corresponding to some element in the ontology) and that are aligned to the ontology (in the sense of sharing the semantic vocabulary). … From the point of view of an application that builds on a given ontology, any semantic representation that is not aligned to the ontology is useless as it could not be processed further by the backend system or application in a meaningful way. (p. 141).

This approach has noteworthy merits: it takes seriously the interrelated needs of ontological modeling and lexical acquisition in service of language understanding; it attempts to foster field-wide acceptance and collaboration by using standard ontologies (e.g., SUMO; Niles & Pease, 2001) and representation formalisms (e.g., OWL, RDF); it uses a modular architecture; and it is the seed of what CU&M envision as "an ecosystem of ontologies and connected lexica, that become available and accessible over the web based on open standards and protocols" (p. 143). CU&M's narrative is grounded in examples from the domain of soccer and offers an accessible developer's view of how to build a knowledge-based system.

On the flip side, as the authors themselves acknowledge, "the instantiation of our approach that we have presented in this book lacks robustness and coverage of linguistic phenomena," which they analyze "not as a deficiency of our approach itself, but of the particular implementation which relies on deterministic parsers, perfect reasoning, and so on" (CU&M, p. 142). They suggest that the answer lies in machine learning, which will be responsible for computing "the most likely interpretation of a given natural language sentence," being trained on ontology-aligned semantic representations (pp. 142–143).

For reasons that will become clear in this book, we find this last bit problematic. That is, CU&M expect machine learning to *somehow* solve what is arguably the biggest problem in all of natural language understanding—lexical disambiguation. Note also that since this approach accommodates only domain-specific inputs and their domain-specific interpretations, it appears to block the possibility of systems learning new things over time, thus forever making human acquisition the only road to system enhancement. In short, although we applaud the direction of this research and its goals, we insert a cautionary statement regarding inflated expectations of machine learning.

To conclude this discussion of ontology, the main thing to remember about intelligent agents is that they cannot be expected to function at a human level without significant domain knowledge. The necessary depth of knowledge cannot be expected to be available for all domains at the same time, any more than we can expect physical robots to be configured to carry out 10,000 useful physical maneuvers at the blink of an eye. This motivates our decision to relegate ontology development to a needs-based enterprise and to work toward enabling agents to acquire this knowledge through lifelong learning by reading and interacting with people. Learning is a multistage process that is addressed in practically all upcoming chapters of the book.

2.8.3 Issues of Lexicon

As with the discussion of ontology, we constrain the discussion of lexicon to issues that are particularly important for agent-building work. For a more comprehensive introduction to lexicon-oriented scholarship, see Pustejovsky and Batiukova (2019).

Issue 1. *Enumerative versus generative lexicons*. An enumerative lexicon explicitly lists all word senses that are to be understood by a language processing system. A generative lexicon (as described, e.g., in Pustejovsky, 1995) encodes fewer senses but associates compositional functions with them, such that certain word meanings can be computed on the fly. In short, a generative lexicon has rules attached to its entries, whereas an enumerative lexicon does not. A big difference? In practical terms, not really.

Anyone who has acquired a lexicon for use in NLU knows that rules play a part. It is highly unlikely that the acquirer of an English lexicon will explicitly list all regular deverbal nouns alongside their verbs (e.g., *hiking* < *hike*) or all regular agentive nouns alongside their verbs (e.g., *eater* < *eat*). At some point during knowledge acquisition or running the language processing engine, lexical and morphological rules expand the word stock to cover highly predictable forms. This process generally does not yield 100% accuracy, and, depending on the application, errors might need to be weeded out by acquirers (for a discussion of the practical implications of using lexical rules at various stages of acquisition and processing, see Onyshkevych, 1997). In short, it would be a rare enumerative lexicon that would not exploit lexical rules at all.

What *is* important about lexical rules, whether they are embedded in an enumerative lexicon or put center stage in a generative lexicon, is that they have to be known in advance. This means that the supposedly novel meanings that generative lexicons seek to cover are actually not novel at all. If they were novel, they would not be covered by generative lexicons either. As Nirenburg and Raskin (2004) suggest,

> a truly novel and creative usage will not have a ready-made generative device for which it is a possible output, and this is precisely what will make this sense novel and creative. Such a usage will present a problem for a generative lexicon, just as it will for an enumerative one or, as a matter of fact, for a human trying to treat creative usage as metaphorical, allusive, ironic, or humorous at text-processing time. The crucial issue here is understanding that no lexicon will cover all the possible senses that words can assume in real usage. (pp. 119–120)

The primary difference between the ultimate (listed or inferred) lexical stocks of, say, Pustejovsky's generative lexicon and the LEIA's lexicon lies in sense "permeation," to use Pustejovsky's term. Pustejovsky argues that a verb like *bake* actually has two meanings—*bring into existence*, as for *cake*, and *heat up*, as for *potato*—and that these senses permeate each other, no matter which one is dominant in a given context. So, for Pustejovsky, *bake a potato* primarily means *heat up* but with a lingering subsense of *bringing into* *existence*. Ontological Semantics, by contrast, rejects this notion of sense permeation on the following grounds: (a) deriving one meaning from the other dynamically is too costly to be worth the effort; it is preferable to list multiple senses and the semantic constraints that support their automatic disambiguation; (b) real language use tends to avoid, not introduce, ambiguity; in fact, speakers generally have a hard time detecting ambiguity even when asked to do so; and (c) we do not see any practical, agent-oriented motivation for introducing a sense-and-a-half situation (Nirenburg & Raskin, 2004, p. 120). The lexicon used by LEIAs reflects a combination of enumerated senses and senses that are dynamically generated as a runtime supplement.

Issue 2. *Manual versus automatic acquisition of lexicon*. Ideally, all static knowledge resources would be acquired either fully automatically or primarily automatically with modest human post-editing. This is, one might say, the holy grail of knowledge-rich NLP. In the late 1980s to early 1990s, NLP centrally concentrated on trying to maximally exploit machine-readable dictionaries that were oriented toward people; however, the results were unsatisfying and the direction of work was ultimately abandoned.[21] As reported by Ide and Véronis (1993) in their survey of research involving machine-readable dictionaries (which bears the suggestive title "Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time?"), "The previous ten or fifteen years of work in the field has produced little more than a handful of limited and imperfect taxonomies." In fact, Ide and Véronis share our group's long-standing belief that it is not impossible to manually build large knowledge bases for NLP, as lexicographers can be trained to do so efficiently. And, ultimately, one needs lexicographers, even in the automatic-acquisition loop, for quality control. The crucial prerequisite for the success of any human-aided machine acquisition is that the output not be too errorful, since that can make the task faced by the human inspector overwhelming. A comparison that will be appreciated by computer programmers is the challenge of debugging someone else's insufficiently commented code.

The LEIA's core lexicon was acquired manually, but, to speed up the process, the acquirers selectively consulted available resources such as human-oriented dictionaries, thesauri, and WordNet. For example, in domains that are of relatively less interest to our current applications, such as types of hats or dogs, we list hyponyms found in WordNet in the lexical senses for *hat* and *dog* without attempting to encode their distinguishing features. Of course, the decision regarding grain size can be changed as applications require, and, ideally, all hyponyms and near synonyms ("plesionyms," to use the term coined by Hirst, 1995) will be distinguished by property values. In addition, we sometimes use the meronymic information in WordNet to jog our memories about words relevant to a domain, and we consult thesauri in order to treat semantic clusters of words at a single go.

The lexicon is acquired in conjunction with the ontology, and decisions about where and how to encode certain types of information are taken with the overall role of both resources in mind. Much of lexical acquisition is carried out in response to lacunae during text processing by the language analyzer. Often, a given text-driven need leads to the simultaneous coverage of a whole nest of words.

Two widespread but ungrounded assumptions (or, perhaps, traces of wishful thinking) about lexicon acquisition are worth mentioning at this point: (a) lexicons will automatically materialize from theories (they won't; lexicons are part of models, not theories)[22] and (b) once the issues discussed in the literature of formal semantics have been handled, the rest of the NLU work will be trivial. In fact, the paucity of discussion of a host of difficult lexical items on scholarly pages is as unfortunate as the overabundance of attention afforded to, say, the grinding rule (the correlation between the words indicating animals and the words indicating their meat) or the interpretation of the English lexeme *kill*. We agree with Wilks et al.'s (1996) critique of the debate over whether *kill* should be represented as CAUSE-TO-DIE or CAUSE-TO-BECOME-NOT-ALIVE:

> The continuing appeal to the above pairs not being fully equivalent (Pulman, 1983a) in the sense of biconditional entailments (true in all conceivable circumstances) has led to endless silliness, from Sampson's (1975) claim that words are "indivisible," so that no explanations of meaning can be given, let alone analytic definitions, and even to Fodor's (1975) use of nonequivalence to found a theory of mental language rich with innate but indefinable concepts like "telephone"! (p. 58).

We make decisions about what to acquire and how deeply to describe given entities based on the usual pragmatic considerations: cost and the needs of applications. Hyponyms of *frog* will remain undifferentiated until we either must differentiate them for a herpetology-related application or we have a cheap way of carrying out the work—as by using semiautomated methods of extracting the properties of different species of frogs from texts (see chapter 7 for a discussion of learning by reading). We have dealt with many difficult lexical issues using the microtheories described in this book, but an important methodological choice in pursuing them is to achieve a sufficient descriptive depth and breadth of coverage while tempering our sometimes overambitious academic enthusiasms.

Issue 3. *General-purpose versus application-specific lexicons*. It is difficult to build useful NLP lexicons without knowing ahead of time what processors or applications they will serve. In fact, Allan Ramsay (cited in Wilks et al., 1996, p. 135) has called this impossible due to the extremely narrow "least common denominator" linking theoretical approaches. The difficulty in finding least common denominators has been met repeatedly in the sister domain of corpus annotation: due to the expense of corpus annotation, effort has been expended to make the results useful for as many practitioners as possible. However, by the time the inventory of markers is limited, on the one hand, by theoretically agreed-upon constructs and labels, and, on the other hand, by the ability of annotators to achieve consistency and consensus, the actual markup is less deep or robust than any individual group or theory would prefer. By committing to a known processing environment, and developing that environment along with the knowledge bases it uses, we have made the endeavor of NLU more feasible.

One way of fully appreciating the advantages of environment specificity is to look at lexicons that were bound by environment-neutral ground rules—as was the case with the SIMPLE project (Lenci et al., 2000). SIMPLE developers were tasked with building 10,000-sense "harmonised" semantic lexicons for twelve European languages with no knowledge of the processors or theoretical frameworks they might need to ultimately serve.[23] For commentary on this difficult task, see McShane et al. (2004).

Issue 4. *The largely language independent lexicon*.[24] Saying that a lexicon is *largely language independent* should raise eyebrows: after all, conventional wisdom has it that whereas ontologies are language independent, lexicons are language dependent. However, it turns out that both parts of this statement require qualification: many if not most resources that are called ontologies these days are language dependent; and at least some computational lexicons have substantial language independent aspects. In the domain of knowledge-rich NLU, the importance of language independent (i.e., crosslinguistically reusable) resources cannot be overstated. Building high-quality resources requires large outlays of human resources, which can be justified if one can significantly reduce the effort of producing equivalent resources in languages beyond the first (for examples, see Nirenburg & McShane, 2009).

In our approach, the most difficult aspect of lexical acquisition—describing the meaning of words and phrases using a large inventory of expressive means—needs to be done only once. The linking of these semantic structures to words and phrases of a given language is far simpler, even though it sometimes requires tweaking property values to convey special semantic nuances. Below we describe some of the expressive means available in a LEIA's lexicon and their implications for a largely language independent lexicon.

Property-modified sem-strucs serve as virtual ontological concepts. The most obvious way to represent lexical meaning in an ontological-semantic environment is to directly map a lexeme to an ontological concept: for example, the canine meaning of the word *dog* maps to the concept DOG. The ontological description of DOG includes the fact that it has all the typical body parts of its parent, CANINE; that it is the AGENT-OF BARK and WAG-TAIL; that it is the THEME-OF both CYNOMANIA (intense enthusiasm for dogs) and CYNOPHOBIA (fear of dogs); and so on. In short, direct ontological mapping does not constitute upper-case semantics in the sense used by logicians because the concept DOG is backed up by a richly informative knowledge structure. In the case of argument-taking lexemes, the syntactic arguments and semantic roles need to be appropriately associated using variables, as shown in our examples of *address* and *see* presented in section 2.3.2.

A variation on the theme of direct concept mapping is to map a lexeme to a concept but, in the lexicon, further specify some property value(s). For example:

- *Zionist* is described as a POLITICAL-ROLE that is further specified as the AGENT-OF a SUPPORT event whose THEME is the NATION that HAS-NAME 'Israel'.
- The verbal sense of *asphalt* (as in *They asphalted our road*) is described as a COVER event whose INSTRUMENT is ASPHALT and whose THEME is ROADWAY-ARTIFACT. The ontological

description of COVER, by contrast, indicates that this concept has much broader applicability, permitting its INSTRUMENT and THEME to be any PHYSICAL-OBJECT.

- The verb *recall*, as used in *They recalled the high chairs*, is described as a RETURN-OBJECT event that is CAUSED-BY a REQUEST-ACTION event whose AGENT is a FOR-PROFIT-CORPORATION and whose THEME is ARTIFACT, INGESTIBLE or MATERIAL. Here, too, the constraints on CAUSED-BY and THEME are narrower than are required by the concept RETURN-OBJECT.

The lexical constraining of ontological property values can be viewed as creating virtual, unlabeled concepts.

The question, then, is how does one determine whether a given meaning should be recorded as a new ontological concept (say, ASPHALT-EVENT) or be expressed lexically by tightening constraints on an existing concept (as in the description of *asphalt* above)? There are two nonconflicting answers to this question. On the one hand, the ontology, as a language-neutral resource, should contain only those meanings that are found in a large number of languages; so whereas DOG is a good candidate for an ontological concept, ASPHALT-EVENT is not. Naturally, this is not a precise criterion, but we do not know how it could be otherwise without embarking on an entire program of study devoted to this decision-making process. On the other hand, for LEIAs, it does not matter whether meanings are described in the ontology or the lexicon since the resources are always leveraged together. So, we could alternatively treat the verbal sense of *asphalt* by creating the ontological concept ASPHALT-EVENT as a child of COVER whose INSTRUMENT is ASPHALT and whose THEME is ROADWAY-ARTIFACT. Then the lexical sense for the verb *asphalt* would include a direct mapping to this concept.

The possibility of creating virtual ontological concepts within the sem-struc zone of lexicon entries is the first bit of evidence that the LEIA's lexicon is not entirely language specific: after all, there are other languages that have lexemes meaning *Zionist*, *to asphalt (some roadway)*, and *to recall (some product)*. Once a sem-struc expressing such a meaning is developed for one language, it can be reused in others.[25]

Alternative semantic representations. Knowledge acquisition requires decision-making at every turn. Consider the multiword expression *weapons of mass destruction*, for which two lexicon acquirers on our team, working separately, came up with different but equally valid descriptions: one was a set containing CHEMICAL-WEAPON and BIOLOGICAL-WEAPON, and the other was WEAPON with the potential (a high value of potential modality) to be the INSTRUMENT of KILLing more than 10,000 HUMANS.[26] These descriptions would both support roughly the same types of reasoning. Since we were not pursuing the domain of warfare in any great detail at the time, these descriptions were created without much ado and either one of them—or a combination of both—aligns with the overall grain size of description of the resources. By contrast, if we had been deeply invested in a warfare-oriented application, we would have sought counsel from a domain expert regarding the best representation of this, and all other relevant, concepts.

The goal for multilingual lexical acquisition is to avoid (a) duplicating the work of analyzing such entities during the acquisition of each new language or, worse yet, (b) endlessly quibbling over which of competing analyses is best. For example, if we had gone with the "KILL > 10,000 HUMANS" approach to describing *weapons of mass destruction*, should the number have been 10,000 or 20,000? What about nonhuman ANIMALS? And PLANTS? Should we have used value ranges instead of single values? The strong methodological preference for reusing semantic descriptions once formulated does not imply that the first lexicon acquired is expected to reflect all the right answers or that the semantic description for that language should be set in stone. However, it does mean that acquirers should choose their semantic battles carefully in order to support practical progress in real time.

Complex semantic descriptions. The more complex a semantic description is, and the more time and thought it takes to create, the greater the benefit of reusing it crosslinguistically. Let us consider just a handful of such examples. Consider the adverb *overboard*, whose lexical sense is shown below.

```
overboard-adv1
   def.   indicates that the source of the motion is a boat and the destination is a body of water
   ex.    They threw the rotten food overboard. He jumped overboard.
   syn-struc
      root    $var1 (cat v)
      mods    $var0
   sem-struc
      ^$var1 (sem MOTION-EVENT)
         SOURCE         SURFACE-WATER-VEHICLE
         DESTINATION    BODY-OF-WATER
```

Overboard is described as modifying a MOTION-EVENT whose SOURCE is SURFACE-WATER-VEHICLE and whose DESTINATION is BODY-OF-WATER. So, if one throws rotten food overboard, THROW fulfills the requirement that \$var1 represent a MOTION-EVENT, and the meaning of *overboard* adds to the TMR an indication of the SOURCE and DESTINATION.

Although this description, once formulated, might seem quite transparent, only an experienced lexicographer is likely to devise it. Moreover, it requires that the acquirer find the three relevant key concepts in the ontology. Together, this is far more work for the first lexicon than replacing the headword (*overboard*) with its equivalent in other languages—such as the prepositional phrase *za bort* in Russian.

Issue 5. *Pleonastics and light verbs.* As we have seen, not every word of a language directly maps to an ontological concept. In fact, many words used in certain constructions do not carry any individual meaning at all. Such is the case, for example, of pleonastic pronouns in examples like *It is snowing (raining)*, *It is thought (known, understood)* *that* ..., and *Someone finds it crazy (strange) that* .... In all these cases, *it* carries no meaning. The best way to record its nonsemantic function is to create multiword lexical senses for all such constructions and, in those sense descriptions, explicitly indicate that *it* is not referential. Formally, we do that using the descriptor *null-sem+*, which means *this element has null semantics*. The meaning of the rest of the construction is expressed in the normal way.

```
rain-v1
    def.    the construction for expressing rain events
    ex.     It is raining. It rains a lot here. It has just rained.
    syn-struc
        subject    $var1 (root it)
        v          $var0
    sem-struc
        RAINSTORM
            ^$var1    null-sem+
```

This construction, like all others, allows for inputs to have different values for mood, tense, and aspect, as well as free modification of all elements that have not been null-semmed.

While an acquirer of a Russian lexicon will not exploit this lexical sense (Russian conveys this meaning using an expression literally translated as *Rain goes*), an acquirer of a French lexicon can, since syntactically and semantically *Il pleut* is equivalent.

Light verbs in multiword expressions are treated similarly. Light verbs are defined as verbs that, at least in certain constructions, carry little meaning, instead relying on their complement noun to provide the meaning. Although English does not use light verbs as widely as, say, Persian, it does have a few, such as *take* in collocations like *take a bath < a shower, a break, a nap, a vacation, a breather >* and *have* in collocations such as *have a fight < an argument, a nice time, a party>*. In addition to these canonical light verbs, some verbs can function as light verbs in certain constructions. For example, there is little difference between saying *someone had a stroke < a heart attack, a seizure>* and *someone suffered a stroke < a heart attack, a seizure >*. The lexical sense for the construction *suffer* + MEDICAL-EVENT is as follows.

```
suffer-v1
    def.    the construction 'to suffer a MEDICAL-EVENT'
    ex.     He suffered a heart attack (a seizure, a stroke).
    syn-struc
        subject         $var1
        v               $var0
        directobject    $var2
    sem-struc
        ^$var2 (sem MEDICAL-EVENT)
            EXPERIENCER    ^$var1
    synonyms    have
```

As this structure shows, the meaning representation is headed by the meaning of the MEDICAL-EVENT that serves as the direct object; the EXPERIENCER of that event is the meaning of the subject of the construction. Note that the function/meaning of *suffer* is folded into the meaning representation. The reason there is no indication of null-sem+ is that it is formally impossible to null-sem the head (i.e., $var0) of a lexical sense since the entire meaning representation expresses the meaning of that string as used in the given construction.

In sum, preparing an analysis system *not* to assign semantics to certain text elements in certain constructions is as important as preparing it *to* assign semantics. The natural place to provide such construction-sensitive information is the lexicon. Since certain types of constructions are prevalent in certain families of languages, sem-strucs that predict the null semantics of given entities can be reused in related languages.

Issue 6. *Tactics for reusing lexical descriptions across languages*. We have already explained that sem-strucs—no matter what lexicon they originate from—represent the meanings of words and phrases in natural languages and that these meanings are largely applicable across languages. The latter derives from the Principle of Practical Effability (Nirenburg & Raskin, 2004), which states that what can be expressed in one language can be expressed in all other languages, using expressive means available in those languages—a word, a phrase, or a lengthy description. That is, we proceed from the assumption that word and phrase meanings across languages are *mostly similar* and that different semantic nuances across languages are an important but less frequent occurrence. Of course, one could also proceed from the assumption that the meanings of words in different languages are *mostly different* and, accordingly, require lexicon acquirers to chase down those differences. The latter approach, while no doubt an enticing opportunity for linguists to demonstrate their language analysis chops, has little place, we believe, in practically oriented tasks—from teaching humans a nonnative language to teaching computers to collaborate with us. In all cases, small nuances of difference should become the focus only long after the basics have been mastered.

All this being said, the job of writing a lexicon for language L2 based on the lexicon for language L1 should, in large part, be limited to providing an L2 translation for the headword(s); making any necessary syn-struc adjustments; and checking/modifying the linking among variables between the syn- and sem-strucs. For example:

- The first noun entry alphabetically in the English lexicon is *aardvark*, which is a simple noun mapping to the concept AARDVARK. If L2 has a word whose meaning corresponds directly to the English word *aardvark* (e.g., Russian has *aardvark*), the acquirer can simply substitute it in the header of the entry.
- The noun *table* has two entries in the English lexicon: a piece of furniture mapping to TABLE, and a structured compilation of information mapping to CHART. The corresponding entries in a Hebrew lexicon will be recorded under two different headwords: *shulhan* and *luah*, respectively.

- Lexical entries for verbs involve more work, mostly because their subcategorization properties must be described. The entry for *sleep* maps to SLEEP and indicates that the subject is realized as the filler of the EXPERIENCER case role. The corresponding entry in the French lexicon will be very similar, with *dormir* substituted for *sleep* in the header of the entry. This is because French, just like English, has intransitive verbs, and *dormir* is intransitive, just like *sleep*.
- If the lexical units realizing the same meaning in L2 and English do not share their subcategorization properties, the acquirer will have to make necessary adjustments. For example, in English the verb *live* meaning *inhabit* indicates the location using a prepositional phrase, whereas in French the location of *habiter* is expressed using a direct object. Even though this slight change to the syn-struc must be entered, this is still much faster than creating the entry from scratch.
- In some cases, a language has two words for a single concept. For example, Russian expresses *man marries woman* using one construction (X_{Male} *zhenitsja na* Y_{Female}) but *woman marries man* using another (X_{Female} *vyxodit zamuzh za* Y_{Male}). In both cases, they map to the concept MARRIAGE, so this simply requires making two lexical senses instead of one.
- The base lexicon might not include a word or phrase needed in the L2 lexicon. In this case, the task is identical to the task of acquiring a base lexicon to begin with.

To test out how realistic bootstrapping our English lexicon to another language would be—knowing from experience that devils lurk in the most unexpected of details—we carried out a small experiment with Polish. While this experiment suggested that a combination of automated and manual bootstrapping would be useful, it also revealed the need for nontrivial programmatic decisions like the following:

- Should the L1 lexicon be treated as fixed (uneditable), or should the L2 acquirer attempt to improve its quality and coverage while building the L2 version? The organizational complexity of working on two or more resources simultaneously is easy to imagine.
- Should L2 acquisition be driven by correspondences in headwords or simply by the content of sem-struc zones? For example, all English senses of *table* will be in one head entry and typically acquired at once. But should all senses of all L2 translations of *table* be handled at once during L2 acquisition, or should the L2 acquirer wait until he or she comes upon sem-strucs that represent the given other meanings of the L2 words?
- To what extent should the regular acquisition process—including ontology supplementation and free-form lexical acquisition—be carried out on L2? Ideally, it should be carried out full-scale for each language acquired with very close communication between acquirers. A high-quality integrated interface that linked all languages with all others would be desirable.

The answers to all these and related questions depend not only on available resources but also on the personal preferences of the actual acquirers of lexicons for given languages working on given projects at given times. Of course, the matter of acquisition time is most pressing for low- and mid-density languages since there tends to be little manpower available to put to the task.

2.8.4 Paraphrase in Natural Language and the Ontological Metalanguage

Our discussion of options for knowledge representation has already touched on paraphrase. Here we delve a bit deeper into that issue, which actually has two parts: linguistic paraphrase and ontological paraphrase.

Natural languages offer rich opportunities for paraphrase. For example, one can refer to an object using a canonical word/phrase (*dog*), a synonymous or nearly synonymous generic formulation (*mutt*, *pooch*, *man's best friend*), a proper name (*Fido*), an explanatory description (*a pet that barks*), or a pointer (*him*). Similarly, one can express a proposition using canonical sentences (*My dog gives me such pleasure*; *I get such pleasure from my dog*; *My dog is such a source of pleasure for me*), special constructions (*My dog, what a source of pleasure!*; *You know what you are, Fido? Pure pleasure!*), and so on. Some of these locutions—such as active and passive voice pairs (*The dog chased the cat / The cat was chased by the dog*)—will generate exactly the same meaning representation. However, even those that do not generate *exactly* the same meaning representation are, in many cases, *functionally* equivalent, meaning that they will serve the same reasoning-oriented ends for the LEIA.[27]

A standing task for LEIAs is determining how newly acquired knowledge correlates with what has previously been learned. Among the simpler eventualities are that the new meaning representation is identical to a representation stored in memory; it is identical except for some metadata value (such as the time stamp); it contains a subset or superset of known properties that unify unproblematically; or it is so completely different from everything known to date that the question of overlap does not arise. None of these eventualities directly involves paraphrase. Two eventualities that *do* involve paraphrase are these: the new meaning representation is related to a stored representation via ontological paraphrase, and the new meaning representation (or a component of it) is related to a stored representation via ontological subsumption, meronymy, or location. We will consider these in turn.

The newly acquired meaning representation is related to a stored memory via ontological paraphrase. Ontological paraphrase occurs when more than one metalanguage representation means the same thing. Ontological paraphrase is difficult to avoid because meaning representations are generated from the lexical senses for words and phrases appearing in the sentence, so the choice of a more or less specific word can lead to different meaning representations. For example, one can report about a trip to London saying *go to London by plane* or *fly to London*, whose respective meaning representations are as follows.[28]

go to London by plane

MOTION-EVENT-7
    DESTINATION   CITY-50
    INSTRUMENT   AIRPLANE-1
CITY-50
    HAS-NAME   'London'

fly to London

AERIAL-MOTION-EVENT-19
    DESTINATION   CITY-50
CITY-50
    HAS-NAME   'London'

Comparing these paraphrases, the word *go* instantiates a MOTION-EVENT, whereas the more specific word, *fly*, instantiates the more specific AERIAL-MOTION-EVENT. However, the *go* paraphrase includes the detail *by plane*, which provides the key property that distinguishes AERIAL-MOTION-EVENT from its parent, MOTION-EVENT, in the LEIA's ontology. That is, the locally specified (not inherited) properties of AERIAL-MOTION-EVENT in the ontology are

AERIAL-MOTION-EVENT
    IS-A   value   MOTION-EVENT
    INSTRUMENT   default   AIRPLANE
        sem   HELICOPTER, BALLOON-TRANSPORTATION

This says that AERIAL-MOTION-EVENT is a child of MOTION-EVENT and that the most common instrument of flying is an AIRPLANE, but HELICOPTER and BALLOON-TRANSPORTATION are possible as well. If the agent uses this information to fill in a property value that was not attested in the input *fly to London*, the resulting representation will be almost identical to the representation for *go to London by plane.*

go to London by plane

MOTION-EVENT-7
    DESTINATION   CITY-50
    INSTRUMENT   AIRPLANE-1
CITY-50
    HAS-NAME   'London'

fly to London

AERIAL-MOTION-EVENT-19
    DESTINATION   CITY-50
    INSTRUMENT   default AIRPLANE
        sem   HELICOPTER, BALLOON-TRANSPORTATION
CITY-50
    HAS-NAME   'London'

The main difference between these representations, which will not affect most types of reasoning, is that the ontologically extracted information is in the form of concepts, not concept instances, so the agent recognizes the information about the instrument as being generic and not part of a remembered TMR. If, in a subsequent utterance, it becomes clear which mode of transportation was used, that specific knowledge would override the generic knowledge posited in the LEIA's memory about this event.

Detecting whether pairs of meaning representations are paraphrases can be accomplished using a fairly simple heuristic: if the events in question are in an immediate (or very close) subsumption relationship, and if the specified properties of the more generic one match the listed ontological properties of the more specific one, then they are likely to be ontological paraphrases. Note that it is not enough for the properties of the events to simply unify—this would lead to too many false positives. Instead, the generic event must be supplied with the property value(s) that define the more specific one. Absent this, the most the LEIA can say is that two events might be related but they are not paraphrases.

The new meaning representation (or a component of it) is related to a stored memory via ontological subsumption, meronymy, or location. When attempting to match new textual input with a stored representation, the question is, *How close do the compared meaning representations have to be in order to be considered a match?* An important consideration when making this judgment is the application in which the system is deployed. In the Maryland Virtual Patient application (see chapter 8), in which virtual patients communicate with system users, sincerity conditions play an important role. That is, virtual patients expect users, who play the role of attending physician, to ask them questions that they can answer; therefore, they try hard to identify the closest memory that will permit them to generate a response.

One foothold for the associated analysis involves the ontological links of subsumption (IS-A), meronymy (HAS-AS-PART), and location (LOCATION). Consider the example of the user (who plays the role of attending physician) asking the virtual patient, *Do you have any discomfort in your esophagus?* The meaning representation for this question is

```
REQUEST-INFO-1
    AGENT               HUMAN-2     ("the physician")
    THEME               MODALITY-1.VALUE
    BENEFICIARY         HUMAN-1     ("the virtual patient")
    MODALITY-1
        TYPE            epistemic
        SCOPE           DISCOMFORT-1
    DISCOMFORT-1
        EXPERIENCER     HUMAN-1     ("the virtual patient")
        LOCATION        ESOPHAGUS-1
    ESOPHAGUS-1
        PART-OF-OBJECT  HUMAN-1     ("the virtual patient")
```

The interrogative mood in the input gives rise to the instance of the speech act REQUEST-INFO, whose THEME is the value of epistemic MODALITY that scopes over the proposition headed by DISCOMFORT-1. Formalism aside, this represents a yes/no question: if the event actually happened, then the value of epistemic MODALITY is 1; if it did not happen, then the value is 0. The event DISCOMFORT-1 is experienced by HUMAN-1, which will be linked to a specific HUMAN (the virtual patient) via reference resolution. The LOCATION of the DISCOMFORT is the ESOPHAGUS of that HUMAN.

If we extract the core meaning of this question, abstracting away from the interrogative elements, we have the meaning for *discomfort in your esophagus*.

DISCOMFORT-1
  EXPERIENCER   HUMAN-1
  LOCATION   **ESOPHAGUS-1**

ESOPHAGUS-1
  PART-OF-OBJECT   HUMAN-1

Let us assume that the virtual patient does not have a memory of discomfort in the esophagus but does have a memory of some *symptom in its chest*.

SYMPTOM-1
  EXPERIENCER   HUMAN-1
  LOCATION   **CHEST-1**

CHEST-1
  PART-OF-HUMAN   HUMAN-1

The components in boldface are the ones that must be matched. Is the LEIA's memory of a SYMPTOM in the CHEST close enough to the question about DISCOMFORT in the ESOPHAGUS? Assuming this LEIA knows what an esophagus is (which is not the case for all LEIAs), then it can recognize that

- DISCOMFORT is a child of SYMPTOM, forming a subsumption link of only one jump;
- ESOPHAGUS has a LOCATION of CHEST—information available in the ontology; and
- both body parts belong to the same human, the virtual patient, as indicated by the PART-OF-OBJECT property.

According to our matching algorithm—in conjunction with the fact that the virtual patient assumes sincerity conditions in its conversations with the physician—the virtual patient's memory of this event sufficiently matches the physician's question, so the virtual patient can respond affirmatively to the question: Yes, it has a memory of the symptom the physician is asking about.

There is one more aspect of ontological paraphrase that deserves mention: the paraphrasing that occurs as a result of a need to interpret signals from perception modes other than language. For example, in the MVP system, virtual patients are capable of interoception, which is interpreting the bodily signals generated by the simulation engine. We will not delve into the details of physiological simulation here (see chapter 8). Suffice it to say that physiological simulations are driven by ontological descriptions that capture how domain experts think about anatomy and physiology, employing concepts such as DYSPHAGIA (difficulty swallowing), PERISTALSIS (wavelike contractions of a series of muscles), and BOLUS (the contents of a single swallow—a chewed piece of food or a gulp of liquid). A given instance of a virtual patient may or may not know about such concepts—it might be endowed with no medical knowledge, some medical knowledge, or extensive knowledge (the latter being true of virtual patients who, by profession, are physicians). It is the job of the interoception engine to align what is generated by the simulation engine—for example, DYSPHAGIA—with the closest available concept(s) in the given patient's ontology, which might be DISCOMFORT in the CHEST after SWALLOW events, or PAIN near the STOMACH when food is stuck there, or various other things. Depending on the level of coverage of a particular agent's ontology, the results of the interpretation of interoception will be different and will constitute ontological paraphrases of the same event.

2.9 Further Exploration

1. Compare the knowledge bases used in OntoAgent with other available online resources:

- WordNet. Read about it at https://wordnet.princeton.edu and explore the resource at http://wordnetweb.princeton.edu/perl/webwn. There are also wordnets for languages other than English.
- FrameNet. Read about it at https://framenet.icsi.berkeley.edu/fndrupal/ and access the frame index—which is the easiest way to explore the resource—at https://framenet.icsi.berkeley.edu/fndrupal/frameIndex.
- Wordnik. Read about it at https://www.wordnik.com/about and search it at https://www.wordnik.com.

2. For each resource, think about/discuss both its potential utility and its limitations for NLP/NLU.

3. We mentioned in passing John McWhorter's take on Neo-Whorfianism. Read his book, *The Language Hoax*, or watch him lecture on the topic on YouTube—for example, at https://www.youtube.com/watch?v=yXBQrz_b-Ng&t=28s.