LLMs as Knowledge Workers

Sergei Nirenburg, Marjorie McShane, Sanjay Oruganti, Jesse English and Christian Arndt
The LEIA Lab, Rensselaer Polytechnic Institute
nirens@rpi.edu, mcsham2@rpi.edu, orugas2@rpi.edu, engli2@rpi.edu, arndtc@rpi.edu

Abstract

Generative AI has a problem with trust. Knowledge-based AI does not, but faces the knowledge bottleneck: the cost of manually building the lexicons and ontologies that enable reasoning, explanation, and targeted correction. We present a neurosymbolic approach toward overcoming this bottleneck. This approach uses a symbolic semantic language interpreter and static knowledge resources (ontology, lexicon, and a corpus of 1,780 validated text meaning representations) of OntoAgent, a cognitive architecture, in coordination with a language model guided by narrative descriptions of decision-making algorithms that encode principles from decades of knowledge acquisition research. The acquisition process: a) is triggered when the semantic anlyzer encounters knowledge gaps while constructing meanng representations; b) proposes new lexicon entries and ontological concepts; and c) automatically validates them against ontological constraints. We report the results of an initial experiment using this approach as the first step toward establishing what an LLM can achieve when boot-strapped from deep knowledge resources and integrated with a knowledge-based analyzer. This work describes a step in an ongoing R&D program whose core objective is to demonstrate that, just as GPUs enabled neural methods to scale, the use of language models as knowledge acquisition tools will enable knowledge-based AI to scale.

Introduction

For AI agents to be trusted as teammates in human-AI teams, they must be transparent and inspectable, capable of causal explanation and commonsense reasoning, and understand their task environment, themselves, and their human collaborators. Such agents require rich world models and true language understanding. The need for these capabilities has been understood since the earliest days of AI. But systems based on logic and symbolic reasoning over formally encoded knowledge required extensive manual knowledge engineering. The need to build knowledge bases was understood as essential, and many efforts were devoted to this task. All made use of available technologies (machine-readable dictionaries, annotated corpora, machine learning techniques for text processing) yet met with only moderate success in overcoming what became known as the knowledge bottleneck. AI turned toward empirical machine learning methods, culminating in today's generative AI paradigm.
Generative AI agents have transformed many tasks, but they cannot serve as repositories of knowledge for cognitive agents. Language models are not trustworthy: they remain fundamentally opaque, and their explanations are post-hoc rationalizations. When an LLM hallucinates, there is no knowledge base or reasoning trace to inspect and correct. But what about using language models as tools to help overcome the knowledge bottleneck? Might this new technology succeed where earlier efforts failed? Might language models enable knowledge-based AI to scale just as GPUs enabled neural methods to scale?

This paper argues that they can—but not in the way most current neurosymbolic approaches propose. The typical strategy retrofits symbolic constraints onto neural cores: as is the norm in the latest Agentic AI approaches, the LLM remains the engine, with knowledge structures as post-hoc filters. A growing number of researchers have recognized the limitations of this configuration and have begun exploring what we call **SyNe** (Symbolic + Neural) architectures, in which symbolic systems play a more central role. In such architectures, symbolic structures define what constitutes valid knowledge, what reasoning operations are permitted, and what explanations count as adequate. Neural components serve as supporting tools by supplying information obtained through distributional analysis methods operating directly over text corpora or over models developed on the basis of textual training data.

Our approach, which we call **OntoAgentic AI**, shares this SyNe orientation, though it differs from other approaches on three issues discussed in the Related Work section below. In a knowledge acquisition configuration presented here,

---

our OntoAgentic AI system uses a symbolic semantic language interpreter to identify gaps in the system's lexicon and learn the corresponding lexicon entries in coordination with a language model. The latter operates over OntoAgent's static resources (ontology, lexicon and a corpus of 1,780 validated text meaning representations). Its operation is explained by narrative algorithm descriptions encoding principles from decades of knowledge acquisition research.

This paper reports on an experiment using a baseline configuration: a language model (Claude 4.5) operating over these resources to learn candidate knowledge items.

The Symbolic Knowledge Resources

Unlike WordNet, FrameNet, or PropBank, which list lexical relations, frame structures, or predicate-argument patterns, OntoAgent's knowledge resources are anchored in an ontological world model and were designed from the outset to support automatic semantic and pragmatic interpretation. We briefly illustrate each resource type. Detailed descriptions of resources can be found in (McShane and Nirenburg 2021; McShane et al. 2024).

The **ontology** has at present ~9,000 concepts organized under three major branches: object, event, and property. Each concept is defined by a set of property-facet-filler triples — on average 16 properties per concept — specifying what is true of the concept and how its values may be used in reasoning (as defaults, as constraints on permissible fillers, as relaxable approximations, etc.). Properties propagate through inheritance: a concept automatically acquires all properties of its ancestors and can add new ones, tighten inherited constraints, or explicitly block them. This means that adding a new concept under, say, DEVICE immediately inherits that devices have positive usable life, are typically made of metal, and can contain physical objects, without any of this needing to be restated. The ontology was built over three decades specifically to support situated natural language processing, and its property structure is what makes automatic semantic validation possible. The **lexicon** contains ~30,000 English word senses mapped to ontological concepts that represent their meaning, potentially with local constraints that further constrain inherited property values. Consider the entry for *bake*:

```
bake-v1
  cat: v
  syn-struc:
     subject:        $var1
     directobject: $var2
  sem-struc:
       PREPARE-FOOD
         agent: ^$var1
         theme: ^$var2 (baked-good)
         instrument: OVEN
```

This entry maps the verb bake to the concept PREPARE-FOOD. The concept brings with it all its inherited property constraints—including AGENT (constrained to HUMAN) and THEME (constrained to FOODSTUFF). Since *bake* is more specific than generic cooking, the entry adds local con-straints that narrow the inherited values: THEME is tight-ened from FOODSTUFF to BAKED-GOOD, and INSTRUMENT is specified as OVEN. This feature allows a single ontological concept to serve multiple related words. Both *cook* and *bake* map to PREPARE-FOOD, but with different constraints on THEME and INSTRUMENT properties.

Meaning representations (MRs) are created by many modules in OntoAgent, including the language analyzer. In this paper, a corpus of validated meaning representations serves as a set of in-context learning data for the LLM. Each MR in this dataset represents the meaning of a sentence using ontological concepts instantiated with specific property values and linked by semantic relations. For example, the sentence *The technician opened the panel* yields the following MR (presented in a much simplified form, with *find-anchor-time a call to a procedure that determines constraints on the time of the event):

```
OPEN-3
  AGENT: HUMAN-14 (SOCIAL-ROLE TECHNICIAN)
  THEME: DEVICE-1
  TIME: *find-anchor-time
```

Allowing the LLMs to use the MR repository as a resource enhances the learning process.

The LLM-Supported Acquisition Approach

In the experiment we describe, knowledge acquisition is triggered in two situations: when the semantic analyzer encounters a **gap** (a word, sense, or construction not in the lexicon) or when the analyzer produces an MR containing **underspecified compositional elements**, signaled by the generic RELATED-TO property in noun-noun compounds.

This acquisition approach draws on two sources of evidence: **knowledge-based** resources (the ontology, lexicon, meaning representations) and **distributional** information (text-level patterns of word usage, collocations, and contextual similarity). Every decision requires evidence from one or both sources, with subsequent validation against ontological constraints.

As an example of knowledge-based decision heuristics, consider the two kinds of knowledge that the above partial MRs contribute to the acquisition task. First, the syntactic structure of the input is determined even in the presence of gaps, which helps to put together the syntactic (syn-struc) zones of the new entries. Second, the partial MRs supply **unilateral selectional restrictions**—constraints that flow from the analyzed context to the gap. If the agent does not know the word *strudel*, after analyzing *He baked a strudel for dessert*, it will identify the unknown word as a noun and create a tentative lexicon entry with BAKED-GOOD as the ontological-semantic anchor in its sem-struc zone because BAKED-GOOD fills the THEME role of PREPARE-FOOD when it is used to express the meaning of the main sense of *bake*.

The acquisition process described here is a rational reconstruction: a post-hoc explanation of the learning steps undertaken by the LLM. We do not claim that this mirrors what the language model actually computes internally. The reconstructed explanations serve three purposes. First, they identify process steps that can be implemented as transparent, rule-based, deterministic procedures that can be verified and debugged (in fact, many of these routines have been implemented in OntoAgent, though not necessarily used for knowledge acquisition). Second, they provide structured training data: pairing inputs with explicit reasoning traces facilitates experimentation with various combinations of neural and symbolic processing, progressively transferring steps to symbolic execution as integration deepens. Third, we plan to use these narrative descriptions as prompts for LLM to generate code for symbolic procedures, which will constitute transparent, inspectable implementations of the heuristic processors these narratives specify. Using LLMs to generate symbolic code from narrative descriptions represents another facet of the SyNe approach: LLMs help not only populate knowledge-based systems but also build the code infrastructure for those systems. The algorithm narrative thus becomes a specification that generates its own implementation.

The algorithms for acquiring lexicon entries and ontological concepts we present below are modular and support implementation of the decision heuristics in either symbolic or neural fashion. Some steps are inherently distributional (e.g., finding words with similar usage patterns); others are inherently symbolic (e.g., checking whether a proposed property value satisfies inherited ontological constraints). In the experiment reported here, the LLM handles all acquisition steps (both those that are inherently distributional and those that are inherently symbolic) but the modular architecture accommodates a variety of hybrid configurations – for example, symbolic modules handling verification while the LLM handles hypothesis generation.

Lexicon Acquisition

When an unknown word is encountered, the system must construct a lexicon entry for it. The part of speech and syntactic information for the target word's entry are derived from the partial MR of the input sentence that triggered the learning algorithm. The core of lexicon entry learning is finding or creating a concept that expresses the meaning of the target word. The process proceeds through two main phases: searching for candidate concepts followed by evaluating whether candidates are adequate.

Candidate concept search attempts to find existing concepts that could represent the target word's meaning. Five search strategies apply in sequence, each capable of succeeding independently:

- *Direct lookup* checks whether a concept named after the word exists (e.g., *hammer* might find HAMMER); this is a weak heuristic because concept naming does not regularly follow English usage.
- *Semantic field expansion* identifies synonyms, hyperonym and hyponyms of the target word using distributional similarity or lexical resources, then checks whether any map to known concepts.
- *Inverse lexical lookup* searches existing sem-struc zones of lexicon entries for concepts whose names match the target word.
- *Distributional similarity* finds words with similar usage patterns and examines what concepts they map to—if *socket, receptacle*, and *bracket* all map to concepts under CONTAINER, this suggests CONTAINER as a candidate.
- *Property-based matching* extracts expected properties from textual context (e.g., if something can be opened and contains objects) and finds concepts with compatible property profiles.

This evaluation follows the **good-enough principle**: if an existing concept adequately supports the inferences required by context and application, no new concept is created, even if the fit is not perfect. This helps to stem ontological proliferation, that is, creating new concepts unnecessarily, which would reduce the reusability and increase maintenance cost.

Once a candidate concept is identified, the system **validates** the proposed mapping against three conditions. First, the candidate concept must exist in the ontology. Second, the concept's properties must support the inferences required by the context. This is where unilateral selectional restrictions contribute: if the partial MR shows INSERT requiring its destination to be a CONTAINER, then whatever *holder* means must be compatible with that role—CONTAINER satisfies this, but EVENT or its descendants would not. The selectional restrictions flow from the analyzed context to the gap, constraining what concepts are admissible. Third, local constraints must be checked against inherited constraints. If the system proposes to narrow the THEME of *bake* from FOODSTUFF to BAKED-FOOD, this is automatically valid because BAKED-FOOD is a descendant of FOODSTUFF. However, local constraints may sometimes contradict inherited ones—the classic example being that birds can fly but penguins cannot. Such contradictions are permitted but flagged for human review, since they may represent legitimate exceptions. Entries that pass all checks are accepted;

those with constraint contradictions are flagged for human validation.

Ontology Acquisition

Ontology acquisition is triggered when no existing concept can adequately represent a word's meaning—that is, when candidate concept search during lexicon acquisition fails. The algorithm creates a new concept and integrates it into the ontological hierarchy.

The first task is to decide on the new concept's place in the ontological hierarchy. Morphological and syntactic information provides a few heuristics: nouns typically map to object-type concepts, verbs to event-type ones; suffixes like *-ness* or *-ity* suggest properties; nominalized forms like *-tion* suggest events; *-er* suggests either humans or instruments. Next, the system examines English definitions attached to existing concepts and checks what concepts distributionally similar words map to. It then compares the target concept's expected properties against property profiles of candidate parents. Candidates are filtered by constraint compatibility: for ELECTRICAL-PANEL, textual evidence in its definition indicates it has fuses as parts, so any candidate parent must permit physical objects as fillers of its HAS-OBJECT-AS-PART property. DEVICE fits the bill – fuses are physical objects – so it "survives." A candidate parent that constrained HAS-OBJECT-AS-PART to FOODSTUFF or LIQUID would be eliminated. When multiple candidates survive, the heuristic favors depth: parents that are lower in the hierarchy provide more specific constraint inheritance, reducing the number of properties the new concept must explicitly specify.

Next, the algorithm must determine how the new concept differs from its parent and siblings. The concept inherits all ancestor properties; the task is identifying which ones need their constraints modified and what additional properties should be added.

For this task, the algorithm draws on corpus evidence processed through the language analyzer. Consider sentences containing *power panel* or synonyms like *breaker box*: *Check that the fuses and circuit breakers on the panel are properly seated* or *Open the panel door before inspection*. When the analyzer processes these sentences, it produces MRs in which the preposition *on* triggers HAS-OBJECT-AS-PART (fuses and circuit breakers are parts of the panel) and panel door similarly identifies DOOR as a part. This corpus-derived, analyzer-processed evidence directly informs property specification: ELECTRICAL-PANEL should have the property HAS-OBJECT-AS-PART filled with CIRCUIT-BREAKER and DOOR.

Dictionary definitions provide complementary evidence. A definition like *a panel housing circuit breakers or fuses for distributing electrical power* confirms the HAS-OBJECT-AS-PART values and suggests a filler for the PURPOSE property (distributing electricity).

Once a candidate concept is selected or a new concept is proposed, the system checks three conditions:

a) no circularity arises in the IS-A chain, and the new concept is not redundant (i.e., it differs from its siblings).

b) any properties included in the description must be ontologically defined for that concept or its ancestors. For example, INSTRUMENT-OF is defined for ARTIFACT and inherited by its descendants; including INSTRUMENT-OF in a concept under ARTIFACT is valid, but including it for any descendant of EVENT would fail the validation.

c) proposed constraints must be checked against inherited constraints. A child concept can tighten inherited constraints (e.g., restricting THEME from PHYSICAL-OBJECT to ARTIFACT) but typically cannot violate them (e.g., specifying THEME as EVENT when the parent requires PHYSICAL-OBJECT). However, exceptions may be legitimate — as with penguins that cannot fly despite being birds. Such contradictions are flagged for human review.

This automatic validation catches errors that would propagate silently in purely neural systems. When the system proposes that ELECTRICAL-PANEL can have CIRCUIT-BREAKERs as parts, it verifies that CIRCUIT-BREAKER is a PHYSICAL-OBJECT (satisfying the constraint inherited from DEVICE). Such validation is truly semantic: checking meaning against formal requirements, not just checking execution or pattern consistency.

The Worked Examples

We demonstrate the acquisition process using the following text from the domain of shipboard maintenance:

Here is how you replace a shipboard power pan-el's fuse. First, verify that the power panel is de-energized. Go to the power panel. Pull out the fuse. Look at the amperage rating on the fuse and remember it. Go to the stores and retrieve a fuse with the same amperage rating from the correct bin. Return to the power panel. Insert the fuse into the holder. Put the old fuse in the discard bin.

Processing was performed by Claude 4.5 operating over OntoAgent's knowledge resources and the partial MR generated by the language interpreter. The interpreter returned a partial MR and flagged 6 unknown words: *amperage, bin, fuse, holder, rating* and *retrieve* and 2 missing word senses: *store-n2* (supply depot) and *pull-v7* (+ out) because the meanings of the existing senses for these words did not match the expected selectional restrictions in the input.

In what follows we trace two examples of lexicon entry acquisition in detail: one that maps to an existing concept with constraints (*holder*), and one that acquires a phrasal entry and triggers the creation of a new ontological concept (*power panel*).

Acquiring the entry for *holder*. The partial MR for *Insert the fuse into the holder* includes an instance of INSERT requiring a value for its DESTINATION property. From this the system derives that whatever *holder* means, it must be compatible with serving as the DESTINATION of INSERT.

The above five strategies for finding a concept fitting the meaning of *holder* are then applied. Direct lookup finds no concept HOLDER in the ontology. Synonym expansion detects *receptacle, socket, mount*, and *bracket*. The system checks the lexicon for these words: *receptacle* maps to CONTAINER; *socket* maps to ELECTRICAL-SOCKET; *bracket* maps to FASTENER. These concepts are added to the candidate set. Inverse lexical lookup searches for the string *holder* in existing lexicon entries and finds no matches. Distributional similarity identifies words with similar syntactic and collocational patterns: *clip, socket, receptacle, bracket*. After filtering out words already found by synonym expansion, the system checks the lexicon for the remaining words and finds that *clip* maps to FASTENER. Property-based matching applies expectations from the partial MR: the target word fills the DESTINATION role of INSERT; the ontology specifies that the filler of DESTINATION must be a descendant of PHYSICAL-OBJECT or PLACE.

Next, the algorithm determines whether CONTAINER adequately represents *holder* against the three conditions defined above. First, CONTAINER does exist in the ontology. Second, its properties support the inferences required by context: CONTAINER can serve as the DESTINATION of INSERT (satisfying the unilateral selectional restriction), and CONTAINER specifies HAS-OBJECT-AS-PART: PHYSICAL-OBJECT, consistent with a fuse being held in place. Third, local constraints are checked: the algorithm proposes adding INSTRUMENT-OF: HOLD to capture that holders specifically function as instruments of holding. This constraint does not contradict any inherited constraints from CONTAINER.

The algorithm concludes that CONTAINER with this local constraint is adequate to represent the meaning of *holder*; no need to introduce a new concept HOLDER. The following entry is then proposed:

```
holder-n1
  syn-struc:
     root holder
     cat n
  sem-struc:
    CONTAINER
           INSTRUMENT-OF: HOLD
```

The proposed entry is subject to the above validation checks: CONTAINER exists in the ontology; INSTRUMENT-OF is defined for ARTIFACT (of which CONTAINER is a descendant); and HOLD is a valid EVENT that can fill the INSTRUMENT-OF property. The entry passes validation.

Acquiring the entry for *power panel*. This example illustrates acquisition triggered not by a gap but by underspecified composition. The analyzer finds entries for both component words: *power* maps to ENERGY (space constraints preclude a description of disambiguation among the three senses of power in the lexicon), and *panel*, though lacking a standalone entry, is listed as a synonym in the entry for *control panel*, allowing the analyzer to construct an on-the-fly entry copying the content of the latter entry's sem-struc: BOARD-FOR-CONSTRUCTION. The resulting MR represents the meaning of the compound *power panel* compositionally:

```
BOARD-FOR-CONSTRUCTION-1
  RELATED-TO: ENERGY
```

The RELATED-TO property signals that the semantic relationship between the compound's components is underspecified. This triggers construction acquisition. The algorithm applies a pointwise mutual information (PMI) test over a text corpus, confirming that *power panel* is a stable collocation warranting a dedicated entry. However, the synonym-based mapping proves inadequate: BOARD-FOR-CONSTRUCTION is defined as a BUILDING-ARTIFACT-PART, but the text explicitly mentions a *shipboard* panel. This mismatch means a concept that better fits the meaning of the target entry is needed.

The above five search strategies are applied. Direct lookup finds no concept for *power panel*. Synonym expansion finds no concepts for *fuse box, breaker box,* or *distribution board*. Distributional similarity and property-based matching identify words that partially match the target's usage profile, resulting in the concepts BOX, ELECTRICAL-EQUIPMENT, CONTAINER and DEVICE added to the candidate set. MR for *de-energize the panel* includes the ELECTRICAL-STATE property; the concepts BOX and CONTAINER do not include it. The MR for *open the panel* requires HAS-OBJECT-AS-PART: DOOR; ELECTRICAL-EQUIPMENT lacks this. DEVICE has dozens of children in the ontology, indicating excessive generality. So, no existing ontological concept works. Ontology acquisition is triggered.

Creating the POWER-PANEL concept. The noun phrase *power panel* indicates an object-type concept. Evidence from textual contexts and the partial MR provides some constraints: it can serve as the DESTINATION of CHANGE-LOCATION (the meaning of *go*), is related to electrical components (*fuses*) and structures (*shipboard*). These properties suggest that the new concept is a descendant of ARTIFACT and related to CONTAINER and ELECTRICAL-EQUIPMENT. The concept is named POWER-PANEL, matching the input construction. Concept names can be refined during human validation.

Parent selection searches for concepts satisfying the derived constraints: descendants of ARTIFACT related to CONTAINER and ELECTRICAL-EQUIPMENT. Three candidates emerge from this search: DEVICE, BUILDING-ARTIFACT-PART (parent of BOARD-FOR-CONSTRUCTION, the original synonym-based mapping), and CONTAINER (that involves HAS-OBJECT-AS-PART). BUILDING-ARTIFACT-PART is eliminated because its definition restricts it to buildings, which clashes with the occurrence of *shipboard* in the text. CONTAINER has no connection to ELECTRICAL-EQUIPMENT, while DEPENDENT-DEVICE (a descendant of DEVICE and the parent of CIRCUIT-BREAKER, the concept in the sem-struc of *fuse*) does. However, the MR for *open the panel* requires HAS-OBJECT-AS-PART: DOOR. Querying the ontology shows that children of DEPENDENT-DEVICE lack this property; DEPENDENT-DEVICE is eliminated as well. DEVICE remains as the only parent candidate.

Property specification draws on the partial MR. For example, *open the panel* has the meaning of *panel* as THEME of OPEN; the concept OPEN requires its THEME to have HAS-OBJECT-AS-PART, so this property is added to POWER-PANEL. *Pull out the fuse* has the meaning of *panel* as SOURCE. *Fuse* maps to CIRCUIT-BREAKER, which becomes the value (filler) of HAS-OBJECT-AS-PART of the new concept. Distributional analysis identifies *distribution panel* as a synonym of *power panel*. Synonyms overlap in meaning, so the concept for *power panel* should incorporate the meaning of *distribution*. In the lexicon, *distribution* maps to DISTRIBUTE, a descendant of EVENT. POWER-PANEL is a descendant of OBJECT. To find how an object can be linked to an event, the system queries the ontology for descendants of the concept PROPERTY with DOMAIN: OBJECT and RANGE: EVENT, and finds the property PURPOSE, yielding PURPOSE: DISTRIBUTE (THEME ELECTRICITY), with THEME constraints inferred from textual context evidence via the lexicon entries for fuse and amperage.

Validation confirms the concept is well-formed: DEVICE exists; the IS-A chain contains no cycles; all property values satisfy ontological constraints (DOOR and CIRCUIT-BREAKER are valid fillers for HAS-OBJECT-AS-PART; DISTRIBUTE is a valid filler for PURPOSE); and the new concept differs from existing descendants of DEVICE. The concept is accepted:

POWER-PANEL
IS-A: DEVICE
HAS-OBJECT-AS-PART: CIRCUIT-BREAKER, DOOR
PURPOSE: DISTRIBUTE (THEME ELECTRICITY)

With POWER-PANEL now in the ontology, the lexicon entry for *power panel* can be completed:

power_panel-n1
cat: n
sem-struc: POWER-PANEL

Summary. The system learned new lexicon entries for six unknown words, two missing word senses, and one nominal compound (*power panel*). Three entries (*bin, rating,* and *store*) mapped directly to existing concepts. Four entries (*holder, fuse, retrieve, pull*) mapped to existing concepts with local constraints. No existing ontological concept was found that adequately described the meaning of two entries (*amperage* and *power panel*). This necessitated the learning of two ontological concepts (POWER-PANEL and AMPERAGE). All proposed entries and newly acquired ontological concepts passed automatic validation.

Related Work

LLM-based knowledge acquisition is becoming a major research focus in what we call the SyNe community. Our approach, OntoAgentic AI, instantiates a particular configuration for a cognitive architecture that uses neural components as supporting tools. This section compares OntoAgentic AI with other SyNe approaches on three dimensions: how acquisition is triggered, what knowledge infrastructure supports it, and how outputs are validated.

Trigger Mechanism. In the approach discussed here, acquisition is triggered by gaps detected by the analyzer while generating MRs or by underspecified compositional elements. These MRs are runtime knowledge. They are generated during acquisition, not merely retrieved from static resources. Neural and symbolic components integrate at multiple stages: the analyzer produces structured representations that constrain and guide LLM processing but some of the modules of the analyzer are implemented neurally.

SOAR and ACT-R process language through pattern matching or LLM queries (Kirk et al. 2024; Wu et al. 2025). They can identify that text contains certain words, but they do not produce structured meaning representations with typed instances and property values. When they encounter an unknown word, they cannot derive what properties that word's meaning must have based on its sentential context, whereas our system knows exactly what it does not know, and the partial analysis constrains what it must learn.

Knowledge Infrastructure. Not all SyNe work focuses on knowledge acquisition. Some approaches aim to make LLMs play the role of cognitive models rather than to acquire new knowledge. Thus, LLM-ACTR (Wu et al. 2025) extracts decision-making traces from ACT-R models, compresses them into latent representations, and injects these into LLM adapter layers. The goal is to make LLMs exhibit more deliberative, human-like reasoning by grounding them in cognitively plausible decision processes. Note the direction of knowledge transfer: LLM-ACTR injects cognitive model traces into LLMs to shape their behavior. Our approach inverts this: we use LLMs to populate and extend symbolic knowledge resources. We focus here on approaches that, like ours, acquire knowledge.

All SyNe approaches recognize that neural and symbolic components have complementary strengths. LLMs excel at pattern recognition, text fluency, and leveraging implicit knowledge from training corpora. Symbolic systems excel at formal reasoning, constraint checking, and maintaining inspectable knowledge structures. The SyNe approaches differ in how to integrate them.

The CoALA framework (Sumers et al. 2024) proposes that control mechanisms from cognitive architectures can be adapted to structure LLM-based agents. The framework borrows from SOAR and ACT-R but remains fundamentally neural: the LLM generates output while the cognitive principles inform, though not control the process.

Several cognitive architectures have begun incorporating LLMs for knowledge acquisition. SOAR's Interactive Task Learning work (Kirk et al. 2024) uses LLMs to extract goal descriptions for specific tasks. This approach analyzes and repairs candidate responses to acquire knowledge consistent with an agent's embodiment and environment. NL2GenSym (Yuan et al. 2025) generates SOAR production rules from natural language, automatically creating procedural knowledge.

Our approach is different in that it operates not over statistical associations among uninterpreted tokens but over content interpreted in terms of an ontological world model that includes structures generated at runtime by a resident language analyzer. As a result, our system learns knowledge that can be inspected, queried, and reasoned over. This infrastructure requires substantial investment to build, but it offers capabilities that other resources lack: inspectability, causal explanation, formal validation, localized correction, and reasoning over ontologically grounded structures. The work we report in this paper aims to reduce the cost of building and extending such infrastructure.

Validation. SyNe systems use a variety of strategies to determine whether acquired knowledge is correct. Some systems validate by checking whether the agent accomplishes its task (Kirk et al. 2024). If a SOAR agent successfully makes coffee using LLM-acquired goal descriptions, the knowledge is deemed adequate. But conflating correctness with fortuitous success is not a reliable strategy: an agent might complete a task despite having incorrect knowledge.

Other systems check execution validity. For example, NL2GenSym (Yuan et al. 2025) validates generated SOAR rules by whether they parse and execute without error. But syntactically valid code can implement wrong behavior. Our approach uses semantic validation: we validate against formal ontological constraints—type restrictions, inher-itance requirements, property compatibility. This catches errors that execution testing would miss. When the system proposes that POWER-PANEL has CIRCUIT-BREAKERS as parts, it verifies that CIRCUIT-BREAKER IS-A PHYSICAL-OBJECT, thus confirming that the constraint inherited from DEVICE is satisfied.

A consequence of combining interpreted knowledge with semantic validation is the ability to correct errors transparently. If the choice of a concept's parent is invalidated, the system reruns parent selection with updated constraints. If a property value violates constraints, the system queries the ontology for valid fillers satisfying the inherited restrictions. Cases that cannot be resolved automatically are flagged for human review. This contrasts with neural systems where errors are distributed across millions of parameters and cannot be isolated and corrected.

Discussion

This experiment was designed as an initial viability test and does not offer a quantitative assessment of efficiency or accuracy at scale. The key finding is that the LLM, when provided OntoAgent's knowledge resources and a partial MR, proposed semantically coherent entries and concepts. In this section we discuss limitations of the approach and future directions.

Benchmarks and quantitative evaluation. Six unknown words and one underspecified compound from a single text demonstrate algorithm coverage across acquisition types: simple lexicon mappings, constrained mappings, and new concept creation. However, the accuracy of the results has not been measured at scale. Rigorous evaluation requires well-developed and task-appropriate benchmarks.

Creating the latter involves using diverse texts across domains, preparing gold-standard versions for acquired entries as well as measures of correctness, coverage and efficiency relative to manual acquisition. Measuring the actual impact on acquisition cost requires comparing time and effort against traditional manual acquisition. Our long-term goal is a workflow where humans serve as validators rather than acquirers, dramatically reducing the cost of knowledge acquisition. This paper establishes that such a workflow is viable: the LLM proposed entries that passed automatic validation, navigated the concept hierarchy appropriately, and handled both gap-triggered and composition-triggered acquisition. Developing benchmarks and quantifying efficiency gains is a priority for future work.

Domain specificity. Shipboard maintenance is a narrow domain with concrete vocabulary. Performance on abstract concepts, metaphorical language, or domains requiring extensive world knowledge remains untested. Our approach relies on a reasonably well-developed base ontology; bootstrapping from minimal resources presents additional challenges. The substantial investment required for creating adequate bootstrapping resources may explain why many

SyNe approaches opt for simpler knowledge resources or purely neural representations.

Despite the above limitations, the experiment demonstrates that the approach works. The LLM successfully navigated the knowledge resources, finding relevant concepts, checking constraints, and proposing coherent entries. Automatic validation caught potential errors. The cascade from lexicon gap to ontology acquisition worked as designed. The entries produced are well-formed and semantically appropriate.

Can LLMs help to code symbolic procedures? The algorithm descriptions presented above have an additional practical application beyond serving as rational reconstructions and training data: they can serve as specifications for LLM-assisted code generation. We have started experimenting with LLMs translating these narrative descriptions into working symbolic code that yields transparent implementations that can be inspected, debugged, and formally verified. We intend to report results in the coming months. This undertaking enables deeper SyNe integration, with LLMs not only performing acquisition tasks but also helping to build the symbolic infrastructure that might eventually replace them for a subset of processing steps. The narrative algorithm specification will yield executable code, and the SyNe approach will extend from knowledge acquisition to include system development tasks.

Conclusion

Generative AI has a trust problem; knowledge-based AI has a cost problem. This paper demonstrates that the two approaches can alleviate each other's weaknesses. By grounding LLM processing in ontologically structured knowledge resources and partial meaning representations, we enable knowledge acquisition that is both efficient and validatable.

The experiment establishes viability. Nine lexicon entries and two ontological concepts were acquired from a short procedural text. All passed automatic validation against formal ontological constraints. The LLM navigated the concept hierarchy and handled both gap-triggered and composition-triggered acquisition.

Three features distinguish this OntoAgentic approach from other SyNe efforts. First, acquisition is triggered by gaps in semantic analysis or underspecified composition, so the system knows precisely what it does not know and partial meaning representations suggest what must be learned. Second, knowledge resources – an ontology with typed properties and inheritance, a lexicon with formal mappings into the ontology, and meaning representations generated at runtime by a resident semantic analyzer – support automatic validation. Third, validation is semantic: proposed entries are checked against ontological constraints, not merely tested for execution or task completion.

The broader significance of the results reported here is in what this approach enables. LLMs excel at pattern recognition, text fluency, and leveraging implicit knowledge from corpora. But they remain opaque, their explanations are post-hoc rationalizations having no direct relation to their explanandum, and their errors are distributed across millions of parameters and cannot be isolated and corrected. Symbolic systems excel at formal reasoning, constraint checking, and maintaining inspectable structures. But building them is currently expensive. By combining the two technologies, OntoAgentic architectures gain the strengths of both: LLMs handle distributional reasoning and hypothesis generation; symbolic processors and knowledge bases provide deterministic conversion from natural language into a semantic representation as well as validation, inspectability, and targeted correction.

The integration demonstrated here is a first step: an LLM guided by symbolic resources and validated against formal constraints. The modular algorithm structure supports progressive transfer: symbolic modules can replace LLM processing for steps where deterministic procedures suffice (constraint checking, inheritance computation), while the LLM continues handling steps requiring distributional knowledge or flexible hypothesis generation. Future work will pursue tighter neurosymbolic integration, symbolic modules (possibly partially or completely generated by an LLM) replacing uninterpreted and opaque LLM processing for a subset of steps, while LLMs will help build the symbolic infrastructure itself and support larger-scale evaluation.

The work we report here is a step in an ongoing R&D program one of whose objectives is to demonstrate that, just as GPUs enabled neural methods to scale, the use of language models as knowledge acquisition tools will enable knowledge-based AI to scale.

Acknowledgments

This research was supported in part by Grants N00014-23-1-2060 and N0014-24-1-2679 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

Kirk, J. R.; Wray, R. E.; and Laird, J. E. 2024. Learning goal definitions interactively from language. Proceedings of the AAAI Conference on Artificial Intelligence, 38(17), 18811-18819.

Lenat, D. B. 1995. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11), 33-38.

McShane, M.; and Nirenburg, S. 2021. Linguistics for the Age of AI. MIT Press.

McShane, M.; Nirenburg, S.; and English, J. 2024. Agents in the Long Game of AI. MIT Press.

Nakos, C.; and Forbus, K. D. 2024. Using large language models in the Companion cognitive architecture: A case study and future prospects. Proceedings of the AAAI Symposium Series, 2(1), 356-359.

Nakos, C.; and Forbus, K. D. 2025. Towards knowledge autonomy in the Companion cognitive architecture. Cognitive Systems Research, 92.

Sumers, T. R.; Yao, S.; Narasimhan, K.; and Griffiths, T. L. 2024. Cognitive architectures for language agents. Transactions on Machine Learning Research.

Wu, S.; Oltramari, A.; Francis, J.; Giles, C. L.; and Ritter, F. E. 2025. Cognitive LLMs: Toward Human-Like Artificial Intelligence by Integrating Cognitive Architectures and Large Language Models for Manufacturing Decision-Making. Neurosymbolic Artificial Intelligence 1. doi:10.1177/29498732251377341

Yuan, L.; Fu, J.; Laird, J. E.; and Prakash, A. 2025. NL2GenSym: Generating SOAR production rules from natural language. arXiv preprint.