Advances in Cognitive Systems 10 (2023) 71–84 Submitted 5/2020; published 5/2023

Content-Centric Computational Cognitive Modeling

Sergei Nirenburg ZAVEDOMO@GMAIL.COM
Marjorie McShane MARGEMC34@GMAIL.COM
Jesse English DRJESSEENGLISH@GMAIL.COM
Language-Endowed Intelligent Agents Lab, Rensselaer Polytechnic Institute, Troy, NY 12180 USA

Abstract

That human-level cognitive systems will require substantial knowledge is commonplace. However, for a number of reasons, issues of large-scale content are not directly addressed by the cognitive systems community. This paper makes the case for the centrality of content for the long-term goals of cognitive system research. Topics addressed include: the nature and scope of content-centric cognitive modeling; its juxtaposition with work on cognitive architectures and cognitive systems; the importance of addressing long-term knowledge needs immediately, since they cannot be met by gradual enhancements in application systems; the ways in which a content-oriented perspective affects the design of cognitive architectures; how content-centric modeling reduces reliance on search; and the practice of content-centric modeling in the OntoAgent environment.

1. Introduction

There is a lacuna at the core of cognitive systems research. One component needed by systems with human-level capabilities is underrepresented. This component is *knowledge content* of a size and sophistication commensurate with the architectures and systems it serves. Modeling human-level functioning will only become possible if cognitive systems are endowed with a substantial amount of knowledge that covers a large inventory of knowledge types described in sufficient detail to support human-level decision making. But knowledge acquisition is a time-consuming undertaking that does not bring about an immediate payoff. So people find ways to avoid or bypass addressing content directly:

- by concentrating on other, equally important, theoretical and engineering issues;
- by selecting application domains where knowledge requirements are limited;
- by claiming that content-related work is outside the purview of cognitive system research and we should import content from outside sources;
- by looking for alternative ways of providing fodder for the heuristics driving cognitive systems, such as machine learning.

The danger of this state of affairs continuing is that, without renewed attention to adequate content, the current crop of cognitive systems will never achieve the desired cognitive functionalities and application scope.

This paper motivates reintroducing broad-coverage content into the purview of cognitive systems research. We discuss the objectives of such work, analyze how taking the content perspective, in turn, influences designs for cognitive architectures, and give examples of the use of this paradigm in our team's past and current research. English and Nirenburg (2020) describe a computational infrastructure for this type of work.

2. The Place of Content-Centric Modeling in the Cognitive Systems Enterprise

The inventory of tasks facing developers of cognitive systems is vast. One direction of work is building cognitive architectures. Langley, Laird, and Rogers (2009) describe nine classes of processes required by such architectures, ranging from recognition and categorization to decision making and learning and communication. Sloman and Scheutz (2002) introduce 13 rubrics or dimensions on which cognitive architectures can be compared. And although no one doubts the importance of actual content (knowledge bases) in cognitive systems, discussions of content are usually at a meta level. For example, Langley et al. focus on its representation, organization, utilization, and acquisition.

Assessing this state of affairs, Lieto, Lebiere, and Oltramari (2018) argue that the major current cognitive architectures are deficient in that they "only process a simplistic amount (and a limited typology) of knowledge." As a result, the authors claim that "the … mechanisms that [cognitive architectures] implement concerning knowledge processing tasks (e.g., retrieval, learning, reasoning) can be only loosely evaluated and compared … to those used by humans." *A fortiori*, today's cognitive systems, by their nature, cannot be expected to help with such evaluation or comparison either – their primary objectives are to achieve a specific well-circumscribed functionality. Lieto et al. recommend that "cognitive architectures should address … the problems concerning the limited size and homogeneity of the encoded knowledge". In support of this recommendation, they describe knowledge-related gaps in some of the most prominent of today's cognitive architectures.

Although Lieto et al.'s criticisms are valid, we believe they are aimed at the wrong target. That is because cognitive architectures, being theoretical constructs, are responsible only for accommodating different types of knowledge, not actually acquiring it. Similarly, architecture researchers should not necessarily be held responsible for developing large knowledge bases. Cognitive architectures can be initially validated using carefully selected subsets of knowledge, along the lines of the broadly-accepted practice in theoretical linguistics of supporting theoretical statements by a small set of carefully selected examples.

However, whereas theoretically-oriented architecture research can be absolved of accounting for realistic-scale knowledge, not so the application systems that build on it. As a rule, such systems use knowledge bases of limited size – just enough to cover the needs of the particular application. This is true even of contributions that are expressly devoted to knowledge (e.g., Jacobsson et al., 2008). The first order of business in developing such systems is to mitigate the complexity of system engineering and promote system performance. In service of these goals, their knowledge bases are constrained not just in size but also in depth of description. Spending resources beyond the immediate needs of an application is imprudent. Efficiency concerns also dictate the preference for strict modularization, with system modules using dedicated knowledge bases.

Some would argue that the current state of affairs is justified not only by immediate engineering needs, but also theoretically. Consider the "now-or-never bottleneck" hypothesis, which Christiansen and Chater (2016) claim to be "a fundamental constraint on language". The now-or-never principle essentially states that all decisions in language processing must be local. Developers of cognitive systems have found this a convenient principle (and slogan).[1] The problem is that the claim

[1] For example, McDonald (1987) followed this principle in the Mumble language generation system.

falls apart when exposed to actual phenomena of natural languages – as convincingly argued in published responses to Christiansen and Chater. To give just a taste: their theory cannot accommodate non-local dependencies, which are a core design feature of natural languages (Levinson, 2016); it does not account for how downstream input affects the interpretation of earlier material, "which shouldn't occur if chunking greedily passes off the early information to the next level" (MacDonald, 2016); it does not account for many pragmatic phenomena that support the communicative function of language, such as clarification, repair, long-distance repetition, and balancing the needs of speakers with those of listeners (Levinson, 2016; Healey et al., 2016; Lewis & Frank, 2016); and their "very bottom-up characterization of chunking is inconsistent with evidence for top-down influences in perception" (MacDonald, 2016).

Clearly, neither these rebuttals nor Christiansen and Chater's proposal can be sufficiently assessed in this short space. However, the main point should be clear. Theories of cognitive functionalities must, from the outset, be informed by a realistic reckoning of the actual scope of eventualities that must be covered. *Awareness of these eventualities is a core aspect of the knowledge that must be recorded in a cognitive system*. If a developer adopts now-or-never as a tenet for its language component, then he or she runs the risk of needing to summarily discard this component and redesign it from scratch if the systems using it are ever to advance beyond toy domains. This is tantamount to building the foundation for a ranch house and expecting to grow it into a skyscraper.

Our insistence that realistic-sized knowledge must be considered from the outset does not mean that we are unsympathetic to extenuating circumstances that lead to small-domain development. Developers of cognitive systems have important engineering concerns – for example near-real-time performance, the need for format conversions between the inputs and outputs of system modules, and achieving satisfactory performance of the individual modules themselves. Indeed, the complexity of integrating the diverse processing modules in a comprehensive cognitive system application is formidable even when knowledge coverage is limited, as in the robotic learning application described by Scheutz et al. (2017). Concerns related to the breadth and depth of knowledge support must be viewed from several perspectives, including extrascientific ones, such as the funding climate. Still, invoking *a priori* questionable simplifications – such as now-or-never language processing – will predictably lead to failure in the face of realistic inputs.

Understanding the demands of realistic-scale knowledge involves one set of concerns, but actually acquiring it involves another. Since knowledge acquisition is expensive, it is natural for cognitive system applications to seek to import whatever knowledge resources are available, as in Forbus' (2018) importation of the OpenCyc ontology. However, experience shows that integrating imported resources is fraught with complications. Not only do they have lacunae; it is the nature of the lacunae that can be their Achilles heel, as Mahesh et al. (1996) discussed with respect to the utility of Cyc for language processing. Also, if undertaken at a large scale, the importation of resources is also quite costly in terms of effort. Our team learned this when we decided to substitute the extensive homegrown suite of pre-semantic processing tools in our language understander, OntoSem, with the resources of the Stanford CoreNLP toolkit (Manning et al., 2014). As described by McShane and Nirenburg (2021, Chapter 3), aligning CoreNLP output with the semantic resources of OntoSem required a major effort.[2]

Before importing resources, it is necessary to assess their usefulness for a system. Too many resources have been created over the past 40 years to mention in this paper. Centrally, they include

[2] This effort was further exacerbated by the frequency of errors (which, for a statistically-trained system, cannot be remedied), and the fact that syntactic parsers are designed to force decisions that, in reality, need to be postponed until semantic analysis can weigh in.

formal descriptions of the world (starting with Cyc and proliferating to many other formal world models) and resources developed as training data for applications of machine learning techniques (e.g., language-oriented resources such as machine-readable dictionaries and annotated text corpora). Not all of these resources have direct utility to cognitive systems. Recommendations concerning the importation of knowledge resources must still be developed.

Of course, importing resources is not the only path forward. A second option is to develop knowledge resources expressly for cognitive systems. This is attractive for content-centric cognitive modeling not only because it involves interesting theoretical and descriptive work but also because it obviates the need to patch over the insufficiencies of imported resources. A third option for acquiring resources is to develop methodologies for combining externally and internally developed resources in ways that assure their smooth interoperability. An alternative to pursuing broad-scale knowledge coverage from the outset is to envisage the gradual acquisition of knowledge over consecutive versions of an application system. However, for typical application systems, which do not employ content-centric modeling, this approach is impractical: experience shows that solutions designed to cover a limited set of examples of a phenomenon do not work when more cases must be covered. Instead, every time coverage is extended beyond the original example set, such solutions are thrown away. For instance, disregarding or drastically reducing the prevalence of lexical ambiguity in language inputs – which is almost universally practiced in current cognitive system applications – obviates a huge requirement for realistic-scale language understanders.[3] As a result, such systems *appear* to perform at a more sophisticated level than could be achieved without such simplifications. It is more efficient – in fact, imperative – to address cognitive phenomena more holistically from the outset, using cognitive modeling.

A key to solving the knowledge problem in cognitive systems is enabling intelligent agents to engage in human-like lifelong learning, which combines language understanding with bootstrapping from existing knowledge bases. However, as early experimentation showed, both the knowledge bases and the language understanding capabilities must have reached a critical level of quality and coverage to support useful learning in this style (Nirenburg et al., 2007).

The content-oriented tasks discussed above have not been a focus of cognitive architecture researchers, as they emphasized the modularization and organization of component processes. They propose a typology and requirements for knowledge components, including heuristics, and suggest formalisms for content and mechanisms for using it. Application systems implement these recommendations. The more theoretically-motivated application systems attempt to carry out as little "tailoring" (to use Forbus' 2018, p. 21, term) of a cognitive architecture as possible, while still striving to meet application requirements. What remains an orphan – undertaken by neither cognitive architecture researchers nor cognitive systems developers – is work on guaranteeing the timely availability and broad coverage of knowledge content, including models of the world, language, and the agent itself – notably, its decision heuristics.

Large-scale and diverse content is a core prerequisite for making cognitive systems truly human-level, as well as for achieving meaningful validation of theoretical statements made in cognitive architecture research. Once the acquisition of content is decoupled from the immediate needs of a particular application, systems will implement not just reactive or preprogrammed agents, but ones

[3] In fact, the field of natural language processing has experimented with many types of simplification over the years, such as forcing people to use controlled languages and pre-editing text as input to machine translation. None of these methods has proven successful, although not for lack of trying.

that expand their capabilities by learning based on what they know. If the cognitive systems community hopes to develop artifacts that emulate human abilities for perception, decision making, and action, then work on content must become more central to the entire enterprise. Statistical and deep learning approaches can provide some grist for the heuristic mill of cognitive systems but, by themselves, do not hold promise for fulfilling this need.[4] If it is neither for cognitive architectures nor for application systems to address the issue, then it follows that the community must promote a new genre of work to bridge this gap.[5]

3. Objectives of Content-Centric Modeling

Before proceeding further, we should clarify what we view as the key objectives of content-centric modeling for cognitive systems. These include a variety of concerns that are related to feasibility and utility, such as:

1. Describing natural phenomena of the world and language, including how they are perceived by agents, in terms of ontological properties and their value sets;
2. Defining types of objects, events, and relations among them, in terms of these properties;
3. Determining how descriptions of *instances* of objects and events are related to descriptions of their *types*;
4. Defining processing paradigms that reconcile efficiency requirements with the necessity of using large amounts of knowledge;
5. Ensuring that modeling supports the specific content needs of different modes of reasoning (cf. Davis & Marcus, 2015, Section 5);
6. Ensuring that the content required to support all components of a cognitive system uses an interoperable metalanguage anchored in a common world model;
7. Using this conceptual infrastructure for ongoing application-oriented knowledge acquisition;
8. Developing methods for automating content acquisition, based on either processing perceptual inputs using available knowledge resources or reasoning over available knowledge;
9. Developing an efficient methodology of converting inputs and outputs of application system modules to and from the interoperable format supporting content-centric modeling;
10. Developing methods for assessing the quality of this acquired content;
11. Developing the infrastructure to store, maintain, and augment this knowledge; and
12. Supporting the integration of content-centric cognitive models of individual phenomena into an overall model of human cognitive functioning.

Some of the above objectives are more difficult to achieve than others. Many can be pursued either as part of work on a particular end application or with the goal of developing generic resources and capabilities to serve a variety of end systems. No matter its emphasis, the utility of content-centric modeling will be in large part assessed by how easily the model can be adapted to new applications. In what follows, we illustrate our views on the nature and utility of such models using examples from our team's research.

[4] Arguing that a new genre is required falls outside the scope of this paper, but Marcus (2020) offers compelling reasons for its necessity.
[5] We distinguish between content-centric modeling as a methodology and content-centric models as concrete instances of such efforts.

To reiterate, Lieto et al. (2018) underscore that insufficient attention to knowledge in cognitive architecture research has impeded progress on cognitive systems. However, not all work in the latter area warrants Lieto et al.'s critique. In fact, our OntoAgent research framework is predicated on the centrality of knowledge, and proof-of-concept systems built within this framework already feature a large *amount* of many *kinds* of knowledge. The core knowledge bases are the ontology, the semantic lexicon, episodic memory, and the agent model. In addition to stored knowledge, agents also generate and use many kinds of data in their normal operation. Tables 1 and 2 illustrate this generalization with respect to language understanding.

This paper does not attempt to cover the actual knowledge content of OntoAgent, much of which McShane and Nirenburg (2021) have described. Tables 1 and 2 illustrate the core principle of content-centric cognitive modeling: *Processing modules can require knowledge from any of the system's knowledge bases and/or data previously generated by any of its modules.* Of course, the opinion that knowledge about the world is required for understanding language is not new. OntoAgent modeling goes further by suggesting that language understanding and general reasoning share the same knowledge substrate. Validating this hypothesis has been an important part of our research agenda that has involved advancing the theory of ontological semantics, developing large-scale knowledge bases and processing models, and configuring proof-of-concept systems that demonstrate the hypothesis' strengths.

4. Parameters and Desiderata for Content-Centric Cognitive Systems

Many issues related to the design of content-centric cognitive systems require further discussion. In what follows, we briefly present our position on how to assess the size of a system's knowledge bases, the types of knowledge it must contain, the grain size of knowledge elements, and the role of search in content-centric cognitive modeling.

4.1 Size Is a Tricky Measure

Lieto et al. state that "…the size problem is intuitively easy to understand (i.e., it concerns the dimension of the knowledge base available to the agents)." Broadly speaking, this is true, but it can lead to misunderstanding because numbers alone do not reflect a resource's utility for an agent system. The current OntoAgent ontology, used by every processing module, covers about 9,000 mostly general-domain concepts but, counted differently, contains over 165,000 RDF triples. The current OntoAgent lexicon for English, used to support language understanding and generation, includes about 30,000 word senses, but this number also hides complexity. For example, the semantic descriptions of most word senses make reference to ontological concepts, but often a simple pointer to a concept is not sufficient. Instead, certain property values of the underlying ontological concepts must be locally modified inside the lexicon entries. This enriches the expressive power of text meaning representations without cluttering the ontological model with excessive, and often language-dependent, concepts. In addition, many lexicon entries record constructions whose components can be fixed or variable, and required or optional, thus covering many shapes of language input. Finally, lexicon entries can contain procedural attachments that specify how to determine meanings in context – such as how to combine *very* with scalar attributes like *tall* and *cold.* In short, simply counting the number of entries in a lexicon, or the number of concepts in an ontology, helps little to assess the coverage provided by a knowledge resource. We must examine the breadth of coverage in combination with the depth of description for each knowledge unit.

Table 1. Knowledge used by the consecutive stages of language understanding in OntoAgent and its source: ontology (O), lexicon (L), dedicated rule sets that support language understanding (R), knowledge components unrelated to language processing (K), and resources aimed at supporting machine learning (ML).

Stage of Analysis	**Content Used**
Pre-Semantic Analysis	– Morphological and syntactic grammars (R) – Gazetteers or ML-based named entity recognizers (R, ML) – Lemmas from the lexicon (L) – Annotated corpora (ML)
Pre-Semantic Integration	– Pre-semantic analysis error recovery rules (R) – Reambiguation rules for syntax-based "decisions" requiring semantic analysis[6] (R) – Rules for linking lexicon senses to words of input (R) – Rules for assessing the confidence of input-to-lexicon linking (R)
Basic Semantic Analysis	– The semantics of the lexical senses of words and phrases (L) – Syntax-to-semantics linking rules derived from lexicon entries (L) – Procedural semantic routines[7] (L) – Ontological constraints to gauge the confidence of semantic dependencies (O) – Microtheories of phenomena (modality, aspect, mood, time, etc.)[8] (L, O, R)
Basic Coreference Resolution	– Select property values of candidate sponsors (O) (see discussion in Section 4.3) – Microtheories for the treatment of specific types of referring expressions: personal pronouns, definite descriptions, etc. (R)
Extended Semantic Analysis	– Object-to-object relations for extra-clausal disambiguation (O) – Microtheories of non-literal language processing (e.g., metaphors) (L, O, R) – Repository of compounding patterns (O, R) – Repository of classes of metonymies (O, R) – Microtheory of extra-clausal disambiguation (O, R) – Idiomatic creativity rule set (R)
Situational Reasoning	– Goal and plan inventory (K) – Agenda of active goals and plans (K) – Models of self and other agents (K) – Microtheories of "mindreading", indirect speech acts, ellipsis, fragments (R, K) – Results of other non-linguistic perception interpretation services (K) – Active concept instances in the situation model (K) – The microtheory of reference (i.e., anchoring referents to agent memory) (K)

[6] Examples include prepositional-phrase attachments, distinguishing prepositions from particles, and the bracketing of nominal compounds comprising three or more nouns.
[7] For example, *very* in *very smart* must dynamically increase the scalar value of INTELLIGENCE used to describe *smart*.
[8] Knowledge to support microtheory implementation is distributed among the ontology, lexicon, and dedicated rule sets.

Table 2. Data that are dynamically generated during language processing and then used downstream. Some results may remain underspecified until a subsequent stage. For a detailed discussion of the stages of language interpretation, see Chapters 3 to 7 of McShane and Nirenburg (2021).

Stage of Analysis	**Content Used**
Pre-Semantic Analysis	– Sentence boundaries – Word information: lemmas, part of speech tags, morphological features – Syntactic parses (constituency and dependency) – Named entities
Pre-Semantic Integration	– Linkings between elements of input and OntoSem lexicon senses – Lexicon entries for unknown words (with underspecified semantics)
Basic Semantic Analysis	– Word-sense disambiguation decisions – Semantic dependency decisions – Identifying the availability of direct and indirect speech-act interpretations – A more precise specification of the semantics of unknown words
Basic Coreference Resolution	– K-best candidate sponsors for textual referring expressions[9] – The sponsor for many elided referring expressions (objects and events) – The understanding that a given referring expression does not require a sponsor – Bridging reference interpretations for definite descriptions
Extended Semantic Analysis	– The semantic/pragmatic incorporation of fragments into the discourse context – A culled set of candidate text meaning representations
Situational Reasoning	– The intended meaning of the input[10] – True reference resolution[11] – Improved interpretation of unknown words – Any residual ambiguities (for optional clarification)

4.2 Heterogeneity

In their critique of cognitive architectures, Lieto et al. argue that a system's world model must include both prototypes and exemplars. OntoAgent incorporates these types of knowledge. Prototypes are encoded in the OntoAgent ontology. For example, the value of the AGENT property of the event MILITARY-OPERATION is specified as any descendant of the concept MILITARY-ROLE on the property's *default* facet, any descendant of HUMAN on its *sem* facet, and any descendant of GEOPOLITICAL-ENTITY on its *relaxable-to* facet. This, among other things, lets the language understander prefer to interpret *The general started the operation* as referring to MILITARY-EVENT and not SURGERY. If the input were *The chief started the operation,* the choice between the two kinds of events

[9] These include personal pronouns, definite descriptions, proper nouns, and demonstrative pronouns.
[10] This includes understanding implicatures, speech-act interpretation, and the like.
[11] This is not textual coreference; it involves grounding all referring expressions in the agent's long-term memory.

Table 3. An excerpt from a script that supports simulating the physiology of an OntoAgent that serves as a virtual patient in a training application for physicians.

```
(gerd
 <…>
 (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))))
 <…>)
```

would require additional situational information because the lexical sense of *chief* is a descendant not of MILITARY-ROLE but of HUMAN, which is the filler of the *sem* facet of the AGENT property of *both* MILITARY-EVENT and SURGERY.

Exemplars are realized as instances of ontological concepts stored in the agent's episodic memory (also referred to in the OntoAgent literature as the *fact repository* or *belief repository*); as such, they are ontologically typed. This means that constraint satisfaction can use static ontological knowledge in addition to the information available in the exemplar. Thus, if Napoleon Bonaparte is listed in an agent's fact repository as an instance of both HEAD-OF-STATE and GENERAL (a descendant of MILITARY-ROLE), then *Napoleon started an operation* will be interpreted as reporting about an instance of MILITARY-EVENT. Of course, if a particular person is remembered as being both a military officer and a physician, the facet-level comparison illustrated above will not resolve the ambiguity, and further analysis will be required – if the agent deems the resolution of this ambiguity essential.

4.3 Depth of Description

Just like people have different amounts of knowledge about specific topics, so do OntoAgents require content describing topics within their application domain to be of finer grain size than general content. The OntoAgent ontology accommodates detailed descriptions of complex events, also called *scripts* (Schank & Abelson 1977). Some scripts are coarse-grained and sketchy. But those describing core processes in an application can become rather fine-grained. Table 3 presents a small excerpt from a script that details the progression of the gastroesophageal reflux disease (GERD) that supports the dynamic simulation of human physiology in the MVP system for training clinicians (McShane & Nirenburg, 2021, chapter 8). The excerpt appears in its original Lisp format and illustrates time management, cause-effect relationships, checking and updating property values, triggering new scripts, and stopping the operation of scripts in progress.

Our choice of how to implement virtual patient simulations carried methodological implications. Physiology is not controlled by cognition (though psychosomatic effects are real enough and were modeled). In principle, we could have developed or imported (had it existed) a statistical, non-explanatory physiological simulation and then built an interpreter to convert its output (such as perceived symptoms) into the uniform OntoAgent metalanguage. This would have made this task similar to interpreting outputs from other perceptual modalities, such as vision or language. However, we decided instead to develop the simulation in the OntoAgent metalanguage from the start, both because having an explanatory model better fit the pedagogical needs of the MVP application and because it eliminated the need for an extra interpreter.

Separately from the ontology, an OntoAgent agent is also endowed with a model of agency and its instances – notably, a model of itself and those of other agents. These models define the agent's and others' goal and plan inventories,[12] as well as their personality traits, biases, emotions, and physical states. An agent typically has several instances of the agency model in long-term memory, one for itself and one for each agent that it knows.[13] As a result, agents are endowed with a theory of mind, which lets them *mindread* other agents.

To illustrate the operation of an agent equipped with models of self and others, consider an example from the MVP application that highlights how virtual patients operating with different content generate different behavior when conversing with a human clinician during a simulated office visit. When the patient presents with gastrointestinal distress, the physician asks, "Have you traveled recently?" A content-poor virtual patient, Tom Dumb, understands the meaning of this input roughly as, "Does your long-term episodic memory contain recent instances of TRAVEL-EVENT with you as the AGENT?". The virtual patient consults its episodic memory, finds just one instance (TRAVEL-EVENT-37) with an appropriate time stamp, and generates the response, "Yes, I went to Philadelphia last Tuesday." Clearly, this is not what the physician had in mind.

By contrast, a content-rich virtual patient, Tom Smart, has several relevant scripts in its ontology – DOCTOR-VISIT, TRAVEL-EVENT, INFECTIOUS-DISEASE – along with a model of the clinician and ontological scripts that enable mindreading. These resources let it carry out a more complex, human-like sequence of operations: (a) establish the intended meaning of the question ("Could travel have caused or contributed to your condition?"); (b) recognize that the physician's goal is to establish a diagnosis; (c) instantiate a subgoal of understanding whether the question is part of a plan toward that goal (rather than, say, small talk, which pursues a different goal); (d) conclude that the question is part of the clinician's diagnosis script, since some kinds of TRAVEL-EVENTs are potential PRECONDITIONs of the event INFECTIOUS-DISEASE; (e) check its episodic long-term memory to see whether it contains any instance of such a TRAVEL-EVENT; (f) establish that this is not the case; and (g) generate a response based on the intended meaning of the question: "No, I haven't traveled anywhere that might have made me sick." The reasoning parts of the above processing sequence involve accessing different types of elements in the agent's memory, which underscores the benefits of the content-centric methodology. Although we developed this example as a proof of concept, our more recent work – within the applications of furniture assembly and driving – attempts to generalize such reasoning in service of dialog-based collaboration and learning.

---

[12] Since OntoAgent agents attempt to model humans, they follow the omnipresent principle of least effort and avoid, if possible, planning from first principles, preferring to adapt and use plans already stored in their memories.

[13] The fact that the agent model of self may differ from what an omniscient observer knows about it provides fertile ground for experimentation on the influence of biased premises in decision making.

4.4 Search and Retrieval

Classical AI research on planning (Ghallab, Nau, & Traverso, 2004) has focused on finding solutions to problems by chaining actions or operators in ways that transform initial states into goal states. This paradigm naturally emphasizes the role of search through a problem space and a common approach to keeping this search tractable relies on heuristics to guide it down promising paths (Newell & Simon, 1976). There is substantial evidence that people resort to heuristic search to solve unfamiliar problems, but the knowledge used remains minimal (Newell & Simon, 1972).

In contrast, everyday problem solving instead relies largely on content-centric methods to handle more routine tasks. These take advantage of solutions to previous problems, or generalizations of them, that are stored in long-term memory. In such cases, the problem solver retrieves promising structures from memory and, if necessary, adapts them to the situation at hand. For example, the shipwrecked Alvaro Cerezo remembered that lenses can start a fire with sunlight, which led him to use a plastic bag filled with sea water for the same purpose.[14] He did not build a plan of action from scratch, but rather amended a known plan to incorporate a nonstandard lens, which required an understanding of lenses.

In summary, content-centric processing replaces search through a problem space with far more efficient "search" through memory, sometimes augmented by adaptation. There has been some AI work on planning that champions this idea (e.g., Veloso & Carbonell, 1993), but it remains relatively rare. This approach to problem solving underscores the central role of knowledge in cognition and downplays the dynamic composition of solutions via search. For this reason, efficient storage and retrieval operations are essential for content-centric approaches that incorporate deep and broad knowledge. This in turn motivates the need for an interoperable metalanguage for representation of content across knowledge sources.

5. Final Thoughts

In making a case for the centrality of content in the long-term goals of cognitive systems research, we have covered a lot of territory. We described the nature and scope of content-centric modeling, juxtaposing it with work on cognitive architectures, on the one hand, and work on cognitive systems, on the other. We explained the importance of addressing long-term knowledge needs immediately, since they cannot be met by gradual enhancements to the knowledge substrate of application systems. We discussed methodologies for acquiring knowledge used in cognitive systems, acknowledging both their inevitable cost and their individual pros and cons. We described how a content-centric perspective affects designs for cognitive architectures. We explained why search efficiency is of limited importance in content-centric modeling, as agents are not expected to create novel plans on the fly and, if faced with an impasse, can to resolve it by interacting with a human or through learning by reading. And we provided evidence that content-centric cognitive modeling has aided development in the OntoAgent environment.

The cognitive systems community adheres to a methodology of research and development that is centered around demonstration systems. This is a venerable approach to testing the maturity of a technology. But its very nature is biased against long-term objectives. Content-centric modeling offers a different and complementary path from theories embodied in cognitive architectures to application systems. In this approach agents continuously accumulate descriptive and decision-

[14] https://www.mirror.co.uk/news/weird-news/real-life-robinson-crusoe-reveals-11976781

supporting knowledge ("lifelong learning") while operating in an application, thus potentially preparing them for use in different applications. This capability supports the long-term goal of shifting efforts from the currently unavoidable handcrafting of knowledge to automatic learning through language understanding, reasoning, and exploration. A more immediate goal is to reach the critical mass and depth of knowledge that will bootstrap lifelong learning in complex domains.

In practice, content-centric cognitive modeling supports reusing knowledge across the set of application systems developed by a research team, so that knowledge for a demonstration system is not thrown away but retained for future use. As a result, the developers of later systems will have less need to create knowledge from scratch. Methodologically, this approach is similar to the aims of the GATE environment for natural language engineering (e.g., Cunningham, Tablan, Roberts, & Bontcheva, 2013). This strategy underscores the importance for content-centric modeling of attention to infrastructure development – not only ergonomic tools for developers and users but also support for efficient knowledge management (storage, update, and retrieval). English and Nirenburg (2020) describe a knowledge management environment designed specifically to support content-centric cognitive modeling.

To sum up, we believe that the content-centric modeling perspective is essential for the long-term success of cognitive systems research. This new paradigm complements work on cognitive architectures and on applications by offering reusable knowledge that can be used in many settings. This content can aid progress both within and across research teams, although the latter will require adjustment of formats. The framework also offers the best hope for developing agents that engage in lifelong learning and, therefore, lower the cost of knowledge acquisition for cognitive systems. Taken together, these features indicate that it should receive substantially more attention from the research community.

Acknowledgements

Many thanks to Pat Langley for many useful comments on a draft of this paper. This research was supported in part by Grants N00014-16-1-2118 and N00014-17-1-2218 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

Barrett, D. P., Bronikowski, S. A, Yu, H, & Siskind, J. M. (2018). Driving under the influence (of language). *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2668–2683.

Christiansen, M. H., & Chater, N. (2016). The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, *39*, 1–72.

Cunningham, H., Tablan, V., Roberts, A., & Bontcheva, K. (2013). Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS Computational Biology*, *9*, e1002854.

Davis, E., & Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, *58*, 92–103.

English, J., & Nirenburg, S. (2020). OntoAgent: Implementing content-centric cognitive models. *Proceedings of the Eighth Annual Conference on Advances in Cognitive Systems*. Palo Alto, CA: Cognitive Systems Foundation.

Forbus, K. D. (2018). *Qualitative representations: How people reason and learn about the continuous world*. Cambridge, MA: MIT Press.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Francisco: Morgan Kaufmann.

Healey, P. G., Howes, C., Hough, J., & Purver, M. (2016). Better late than Now-or-Never: The case of interactive repair phenomena. *Behavioral and Brain Sciences*, *39*, e76. https://doi.org/10.1017/S0140525X15000813

Jacobsson, H., Kruijff, G-J., Hawes, N. & Wyatt, J. (2008). Crossmodal content binding in information-processing architectures. *Proceedings of the Third ACM/IEEE International Conference on Human-Robot Interaction* (pp. 81–88). Amsterdam: ACM Press.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141–160.

Langley, P. (2018). Theories and models in cognitive systems research. *Advances in Cognitive Systems*, *6*, 3–16.

Levinson S. C. (2016). "Process and perish" or multiple buffers with push-down stacks? *Behavioral and Brain Sciences*, *39*, e81. https://doi.org/10.1017/S0140525X15000862

Lewis, M. L., & Frank, M. C. (2016). Linguistic structure emerges through the interaction of memory constraints and communicative pressures. *Behavioral and Brain Sciences*, *39*, e82. https://doi.org/10.1017/S0140525X15000874

Lieto, A., Lebiere, C., & Oltramari, A. (2018). The knowledge level in cognitive architectures: Current limitations and possible developments. *Cognitive Systems Research*, *48*, 39–55.

MacDonald, M. C. (2016). Memory limitations and chunking are variable and cannot explain language structure. *Behavioral and Brain Sciences*, *39*, e84. https://doi.org/10.1017/S0140525X15000898

Mahesh, K., Nirenburg, S., Farwell, D, & Cowie, J. (1996). *An assessment of CYC for natural language processing*. Technical Report, Computing Research Lab, New Mexico State University. https://www.researchgate.net/publication/277296502_An_Assessment_of_CYC_for_Natural_Language_Processing

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55–60). Stroudsburg, PA: Association for Computational Linguistics.

Marcus, G. (2020). The next decade in AI: Four steps towards robust artificial intelligence. arXiv: 2002.06177.

McDonald, D. D. (1987). Natural language generation: Complexities and techniques. In S. Nirenburg (Ed.), *Machine translation: Theoretical and methodological issues*. Cambridge, UK: Cambridge University Press.

McShane, M., & Nirenburg, S. (2021). *Linguistics for the age of AI*. Cambridge, MA: MIT Press.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Newell, A. & Simon, H. A. (1976). Computer science as empirical enquiry: Symbols and search. *Communications of the ACM*, *19*, 113–126.

Nirenburg, S., Oates, T., & English, J. (2007). Learning by reading by learning to read. *Proceedings of the International Conference on Semantic Computing*. Irvine, CA: IEEE.

Schank, R., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ: Lawrence Erlbaum.

Scheutz, M., Krause, E., Oosterveld, B., Frasca, T., & Platt, R. (2017). Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. *Proceedings of the Sixteenth Conference on Autonomous Agents and Multiagent Systems* (pp. 1378–1386). Sao Paulo, Brazil: ACM Press.

Sloman, A., & Scheutz, M. (2002). A framework for comparing agent architectures. *Proceedings of the 2002 UK Workshop on Computational Intelligence*. Birmingham, UK.

Veloso, M. M., & Carbonell, J. G. (1993). Derivational analogy in Prodigy: Automating case acquisition, storage, and utilization. *Machine Learning*, *10*, 249–278.