Decision-Making During Language Understanding by Intelligent Agents

Marjorie McShane and Sergei Nirenburg

Rensselaer Polytechnic Institute, Troy, NY, 12180, USA
{mcsham2, nirens}@rpi.edu

Abstract. In cognitive modeling and intelligent agent design, a widely accepted architectural pipeline is *Perception–Reasoning–Action*. But language understanding, while a type of perception, involves many types of reasoning, and can even involve action, such as asking a clarification question about the intended meaning of an utterance. In the field of natural language processing, for its part, the common progression of processing modules is *Syntax–Semantics–Pragmatics*. But this modularization lacks cognitive plausibility and misses opportunities to enhance efficiency through the timely application of knowledge from multiple sources. This paper provides a high-level description of semantically-deep, reasoning-rich language processing in the OntoAgent cognitive agent environment, which illustrates the practical gains of moving away from a strict adherence to traditional modularization and pipeline architectures.

Keywords: natural language understanding, intelligent agents, reasoning, cognitive architecture

1 Introduction

The analytic method in science prescribes decomposing problems into subproblems, finding solutions to those subproblems, then synthesizing the solutions. Despite the well-known benefits of such modularization, it has certain unfortunate consequences that have come center stage in our work on developing the cognitively modeled agents we call OntoAgents. Strict modularization of perception, reasoning and action fails to capture the rich information transfer that appears to characterize human cognition and behavior. Our current work on OntoAgents attempts to more accurately model general artificial intelligence by integrating these cognitive modules. In this paper, we discuss one aspect of this integration: *the integration of decision-making (traditionally subsumed under reasoning) into the process of natural language understanding (traditionally subsumed under perception).*

OntoAgents feature integrated physiological and cognitive simulations, modeling the body and the mind. The mind-body connection is modeled as the process of interoception, i.e., the perception of bodily signals [5], [13]. To date, the simulated minds of implemented OntoAgents have shown the capabilities of goal-oriented planning, decision-making influenced by personal biases and

situational parameters, learning, memory management, and natural language processing (see [7], [8], [6], among others).

In this paper we present a conceptual overview of our work toward transcending the boundaries of processing modules in models of cognitive agency. Our effort addresses two separate modularizations – the traditional *Perception–Reasoning–Action* pipeline of cognitive architectures and the familiar *Syntax–Semantics–Pragmatics* pipeline of AI-oriented natural language processing.

Pipeline-oriented approaches, while differing in many respects, typically share the following two characteristics: a) the processing of an input by any module can start only after the upstream modules have finished with this input; and b) the machinery and knowledge resources of each module are typically opaque to those of other modules. There are engineering-oriented reasons for imposing these constraints. But we hypothesize that they are not optimal either as features of cognitive models or as architectural choices in computational implementations of cognitive models.

Issues of modularity and computational architectures have been amply debated in cognitive science and artificial intelligence. This paper is not meant as a contribution to those debates. Our specific objective is to enhance the efficiency and effectiveness of artificial intelligent agents by improving the ways in which they apply knowledge. This objective complements rather than competes with work on enhancing the functioning of agents through more sophisticated formalisms and improved algorithmic efficiency.

We believe that moving away from pipelines will increase verisimilitude in modeling human behavior. In this respect, we are motivated by two working hypotheses. (1) The *inclusivity hypothesis* suggests that cognitive agents, at any given time in their functioning, can apply any and all heuristics currently available to them, irrespective of the provenance of those heuristics. (2) The *least effort hypothesis* motivates agents, in well-defined aspects of their functioning, to "jump to conclusions" – i.e., to declare their current task completed and avoid exhaustive processing. Such decisions are a function of the agents' knowledge and beliefs, their personality traits, and situational constraints. This hypothesis is observationally quite plausible, as anybody who has ever been justifiably interrupted in a dialog can attest (i.e., if the interlocutor has already understood one's point well enough to respond, interrupting can be appropriate).

2 Issues with Pipelines

One insufficiency of the *Perception–Reasoning–Action* pipeline is that it obscures the fact that language understanding, a type of perception, itself routinely involves reasoning and action. Such tasks as lexical and referential disambiguation, the detection and reconstruction of elliptical gaps, and the understanding of indirect speech acts are reasoning-intensive. Moreover, if an agent is intended to model human performance, it must be able to look beyond the boundaries of the narrowly defined language understanding task to judge its confidence in the results of its language processing. If, by the time it finishes processing a language

input, the agent is confident that it has understood the input, this should lead to reasoning and action. If, by contrast, the agent has not sufficiently understood the input, then it must select a recovery strategy. One such strategy is the *action* of asking its human collaborator for clarification. Incorporating such reasoning and action into the perception module, we arrive at the following, more realistic, workflow, in which parentheses show optionality: *Perception and reasoning about perception–(Reasoning about suboptimal perception processing–Recovery action)–Reasoning–Action.*

With respect to language modeling itself, the traditional, theory-driven *Syntax–Semantics–Pragmatics* pipeline fails to accommodate the large number of cross-modular methods available for treating individual linguistic phenomena. To take just one example, many instances of ellipsis – the null referring expression – can be detected and resolved *prior to* semantic analysis, with the results then being available to inform semantic analysis.[1] Therefore, just as we modified the cognitive modeling pipeline above, so must we modify the language processing pipeline, leading to the more functionally sufficient approach detailed in Section 4.

3 Pursuing Actionable Language Analyses

The goal of language understanding in OntoAgent is for the agent to arrive at an *actionable* interpretation of text input. We define as actionable those interpretations that are deemed by the agent to be sufficient to support post-perception reasoning and action. An actionable interpretation might represent a complete and correct analysis of all input strings, or it might be incomplete; it might involve only a partial analysis of the input strings, or it might invoke maximally deep reasoning; and it might be achievable by the agent alone, or it might require interactive clarifications or corrections by a human or artificial collaborator. In short, for each language input, after each stage of processing, the agent must estimate whether it has arrived at a level of input understanding sufficient for passing control to the reasoning and action modules. As soon as the answer is positive, it can proceed to post-perception reasoning and action.

This modeling strategy reflects our belief that, in order to foster the development of viable agent applications at a time when the state of the art cannot yet support full and perfect semantic analysis of unrestricted input, it is necessary to define practical halting conditions for language analysis. Consider an example from an OntoAgent prototype system called Maryland Virtual Patient [5], [13]. One of the intelligent agents in this system plays the role of a virtual patient being diagnosed and treated by a human medical trainee. During simulated office visits, the virtual patient engages in dialog with the trainee during which the latter can ask questions, suggest diagnostic and treatment protocols, provide background knowledge about the patient's disease, and answer the patient's questions. In each of the trainee's dialog turns, the agent attempts to

[1] If ellipsis were to be treated like other referring expressions, it would normally be subsumed under pragmatic analysis.

detect something actionable, such as a question it should answer or a recommendation it should respond to. Responding to this actionable input becomes the agent's communicative goal of choice, absolving it from the necessity of full and confident analysis of every element of input.

This type of incomplete processing is not merely an escape hatch for modeling intelligent agents in the early 21st century. We believe that it models how people naturally behave in communicative situations: they pay attention to the main point but often ignore many of the details of what others say. For example, if a doctor provides exhaustive detail about the potential side effects of a medication, do live patients pay full attention? Would they understand and remember every detail even if they did? Selective attention is a manifestation of the principle of least effort; it represents natural conservation of energy and thus protects against cognitive overload [15]. So, even though OntoAgents show "focused attention" for practical reasons, the effects of this behavior in simulation will, we hypothesize, make agents more human-like.

We will now consider, in turn, how the canonical pipelines introduced above can be modified to better serve OntoAgents in their quest for actionable language interpretations.

4 The Stages of Language Analysis

To reiterate, the agent's goal in processing language input is to arrive at a confident, actionable analysis as soon as possible. For this reason, we are working toward configuring agents that can treat phenomena as soon as the necessary heuristic evidence becomes available. At any point in language analysis, an agent should be able to decide that the current state of analysis is actionable and proceed directly to post-perception reasoning and action. We discuss the stages of language processing under development in the order presented below.

1. Perception and reasoning about perception
   (a) Exploiting situational expectations and conventions
   (b) Syntactic analysis
      i. Syntactically-informed reference resolution
      ii. Tree trimming
   (c) Semantic analysis
      i. Semantically-informed reference resolution
      ii. Semantically-informed speech act understanding
   (d) Reference resolution
   (e) Indirect speech act interpretation
   (f) Reasoning about suboptimal perception processing
      i. Recovery action
2. Post-Perception Reasoning
3. Action

Space constraints preclude a detailed description of *how* the system arrives at each type of analysis or a detailed rundown of results to date. Regarding the latter, we have recently evaluated our engines for basic semantic analysis [6], the treatment of multi-word expressions [12], verb phrase ellipsis resolution [10] and tree trimming in support of the latter [11]. The other microtheories mentioned above are at various stages of development. The rationale behind presenting this blueprint for agent functioning even before an end-to-end implementation is available is that we believe that drawing the big picture is an essential prerequisite for long-term progress on the many component challenges of configuring truly intelligent artificial agents. The modest goal of the current contribution is to motivate the reconceptualization of the traditional pipeline architectures introduced earlier.

1a. Exploiting Situational Expectations and Conventions The first stage of language processing relies on textual string matching. The hypothesis is that some combinations of strings – which can even be entire sentences – are so frequent or expected that they are stored in memory along with their semantic analyses, thus not requiring compositional analysis at each encounter. For example, in the Maryland Virtual Patient application, we stored semantic analyses of expected formulaic inputs such as *How are you feeling?* Storing remembered analyses not only speeds up system functioning and reduces unexpected misinterpretations, it also reflects the human-oriented hypothesis that, in accordance with the principle of least effort, people store frequently encountered phrases as ready-made information bundles.

1b. Syntactic Analysis. If the agent does not treat an input "reflexively", it proceeds to syntactic analysis. Stanford CoreNLP [4] provides tokenization, sentence splitting, PoS tagging, morphological analysis, named entity recognition, syntactic immediate constituent analysis and a dependency parse. Although syntactic analysis represents only an intermediate result toward semantic analysis, it can inform certain types of decision-making. For example, an agent might choose to further process sentences only if they contain certain keywords, or combinations of keywords, of interest.

1bi. Syntactically-informed reference resolution. Next the agent engages in a series of reference resolution procedures that are undertaken at this early stage because they require as input only the results of syntactic analysis and access to the lexicon. For example, our agents can detect and resolve verb phrase ellipsis in sentences like *They attempted to* ***win the tournament*** *but couldn't [e]*, as described in [10]. Similarly, they can establish lexico-syntactically-based coreference links for a pronominal referring expressions in certain linguistically defined configurations.

The benefits of early reference processing cannot be overstated. Detecting ellipsis and reconstructing the missing string permits the meaning of the expression to be computed during basic semantic analysis. Continuing with the example from above, the agent will actually be semantically analyzing *[They]-1 attempted to [win the tournament]-2 but [they]-1 couldn't [win the tournament]-2*, in which the indices indicate coreference. Similarly, establishing high-confidence textual

coreference relations for overt pronouns at this stage enhances the simultaneous disambiguation of those expressions and their selecting heads. For example, it is much easier for the agent to disambiguate both the subject and the verb in *The train stopped* than to disambiguate these strings in *It stopped*. So, coreferring *it* with *train* in a context like *The train raced toward the station then it suddenly stopped* is of great benefit to semantic analysis.

1bi. Tree Trimming Before proceeding to semantic analysis, the agent has the option of carrying out "tree trimming," also known as syntactic pruning or sentence simplification. Tree trimming refers to automatically deleting non-core syntactic structures, such as relative clauses and various types of modification, so that the core elements can be more effectively treated.[2] It has been used in applications ranging from summarization to information extraction to subtitling. An agent's decision about whether or not to trim should be a function of (a) sentence length, (b) the constituents in the parse tree and the dependency parse, and (c) situational non-linguistic parameters, such as the agent's cognitive load and the importance of the goal being pursued through the communication.

1c. Semantic Analysis. Semantic analysis in OntoAgent is defined as generating an ontologically-grounded text meaning representation (TMR) that includes the results of lexical disambiguation and semantic dependency determination.[3] TMRs are written in a metalanguage they share with the ontology and other knowledge repositories in OntoAgent. For example, the TMR for the input *Dr. Jones diagnosed the patient* is shown in Table 1. Small caps indicate ontological concepts and numerical suffixes indicate their instances. The "textstring" and "from-sense" slots are metadata used for system debugging.

Table 1. TMR for *Dr. Jones diagnosed the patient.*

DIAGNOSE-1		
	AGENT	HUMAN-1
	THEME	MEDICAL-PATIENT-1
	TIME	(before find-anchor-time) ; indicates past tense
	textstring	"diagnosed"
	from-sense	diagnosed-v1
HUMAN-1		
	AGENT-OF	DIAGNOSE-1
	HAS-NAME	"Dr. Jones"
	textstring	"Dr. Jones"
	from-sense	*personal-name*
MEDICAL-PATIENT-1		
	THEME-OF	DIAGNOSE-1
	textstring	"patient"
	from-sense	patient-n1

[2] For our approach to tree trimming in service of ellipsis resolution see [11].
[3] The OntoSem process of semantic analysis is described in [6] and [14].

Every TMR produced by an agent is assigned a confidence level, which reflects the extent to which lexically and ontologically recorded expectations resulted in a single, unique analysis of the input. The more instances of residual ambiguity, the lower the overall confidence.

Although this example sketches the basic idea of semantic analysis in OntoAgent, it fails to convey that this stage of processing actually incorporates some aspects of early pragmatic analysis. For example, TMRs include the results of reference processing carried out earlier (cf. 1bi above). They also may include newly computed aspects of reference resolution as well as the treatment of indirect speech acts. We consider each of these in turn.

1ci. Semantically-informed reference resolution. The OntoSem lexicon contains lexical senses that support the detection of certain kinds of ellipsis and the resolution of certain kinds of overt referring expressions. For example, there is a sense of the verb *start* that expects its complement to be an ontological OBJECT rather than an EVENT, as in *She started the book*. This sense asserts that there is an elided event whose meaning the system should attempt to recover from the context – all during this same pass of basic semantic analysis. Other referring expressions that are treated using lexically-recorded procedural semantic routines are indexicals such as *yesterday* [9].

1cii. Semantically-informed speech act understanding. The OntoSem lexicon includes a broad range of phrasal constructions that help to reduce the ambiguity of compositional semantic analysis [12]. Among these constructions are conventionalized speech acts. For example, *Could you please tell me X* is interpreted as REQUEST-INFO THEME [the meaning of X]; *I would recommend X* is interpreted as REQUEST-ACTION [the meaning of X]; and so on. Rather than postpone indirect speech-act detection until the downstream module dedicated specifically to it, our system analyzes the semantics and the pragmatics of conventionalized indirect speech acts simultaneously.

This "Semantic Analysis" level of processing will not yet be actionable for intelligent agent applications since referring expressions have not yet been anchored in memory. However, for non-agent-oriented NLP applications, this level of output could be useful since lexical disambiguation has been carried out, the semantic dependency structure has been established, many textual *co*reference relations have been resolved, and some indirect speech acts have been detected.

1d. Reference resolution. Unlike reference resolution procedures undertaken up to this point, OntoAgent's nascent reference module (a) will have access to full semantic analysis as input, (b) will attempt ontology-based reasoning, if needed, and (c) will posit as the goal not just detecting textual *co*reference, but carrying out concept-level reference resolution, which will result in anchoring referring expressions to concept instances in agent memory. For example, given an input like *He began operating on the patient at 7 a.m.*, the system might have several males in the preceding context that could plausibly be the sponsor for the referring expression *he*. However, it is likely that only one of them is listed in the agent's fact repository with the property-value pair SOCIAL-ROLE SURGEON. The key to selecting the correct sponsor is consulting the ontology and

determining that the AGENT of the ontological concept (event) SURGERY – which was activated as the contextually appropriate meaning of *operate* – is typically a SURGEON. This is an example of "reasoning about perception." Note that if earlier reference processing had resulted in textual coreference links, true reference resolution to agent memory would still have to be undertaken at this stage. This would happen, for example, given the input, *After the surgeon completed the surgery, he changed into street clothes*. Here, the grammatical structure strongly suggests the coreference relationship between *he* and *the surgeon*, but this chain of coreference must still be anchored to the right instance of SURGEON in agent memory.

1e. Indirect speech act interpretation. In its current state, our microtheory of non-lexically-supported speech act interpretation covers exclusively application-specific cases. For example, in the MVP application, if the input includes reference to a symptom, but the input overall is not recognized as an instance of asking whether the patient is experiencing that symptom, the patient nevertheless responds as if it had been asked that question. Work is underway to extend this microtheory to cover more generic contexts.

By the time the agent reaches this point in language analysis, it will have carried out all of its basic analysis processes, constructed a TMR, and grounded concept instances in memory. Its overall analysis is associated with a cumulative confidence value that is computed as a function of its confidence about every component decision it has made: each instance of lexical disambiguation, each instance of reference resolution, etc. If the agent's overall confidence is above a threshold, the analysis is declared to be actionable. If not, the agent must decide how to proceed.

1f. Reasoning about suboptimal perception processing. If the agent chose earlier not to carry out syntactic trimming, it can choose to invoke it at this point, in hopes of being able to generate a higher-confidence TMR from a less complex input. The sequence *syntactic analysis – semantic analysis – tree trimming – semantic analysis* is another example of interleaving modules of processing beyond the rather simplistic original pipeline. If the trimming strategy is either not available (e.g., it has been carried out already) or is not favored by the agent (e.g., this is a high-risk situation with no room for error), the agent can undertake a recovery action.

1fi. Recovery action. If the agent is collaborating with a human, one recovery option is to ask a clarification question. This is particularly well-motivated in high-risk and/or time-sensitive situations. There are, however, other options as well. For example, if the analysis problem was due to "unexpected input" – e.g., an unknown word – the system can attempt learning by reading, as described in [2]. Or, the agent can decide to recover passively, by not responding and waiting for its interlocutor's next move which, in some cases, might involve linguistic clarifications, restatements, etc.

2. Post-perception reasoning & 3. Action. These modules of agent cognition take as input whatever results of language processing the agent considered an appropriate stopping condition.

5 Final Thoughts

The recognition that reasoning is needed for language processing is, of course, not novel. The idea has been addressed and debated from the early days of AI-NLP and cognitive science in works by Schank [16], Wilks [17], Woods [18], and many others. Our contribution is an attempt (a) to integrate a larger inventory of more detailed explanatory models that rely on broader and deeper knowledge bases, and (b) to arm agents with the ability to reason about their confidence in language processing and act accordingly. In this regard, it is noteworthy that a central contributor to the success of the Watson system in the *Jeopardy!* challenge was its use of confidence metrics in deciding whether or not to respond to questions [3].

The idea of interleaving processing stages is also not unknown in computational linguistics proper. For example, Agirre et al. [1] use semantic information to help determine prepositional phrase attachment, which is required for producing the correct output of syntactic analysis. Our work differs from contributions of this kind in that our ultimate goal is not success of a particular stage of language processing but, rather, deriving the semantic and discourse/pragmatic meaning of the input using all available clues.

In this space, we were able to give only a high-level overview of language understanding in OntoAgent, along with our methods of incorporating reasoning and decision-making into the process. Naturally, many aspects of this vision of agent functioning are work in progress. Our practical results, which vary across microtheories, have been reported in the cited literature. Near-term goals include both further developing the theoretical substrate of OntoAgent – continuing the genre of the current contribution – and increasing the breadth of coverage of all of the microtheories, knowledge bases and processors that contribute to the functioning of OntoAgents.

Acknowledgments This research was supported in part by Grant N00014-09-1-1029 from the U.S. Office of Naval Research. All opinions and findings expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

1. Agirre, E., Baldwin,T., Martinez, D.: Improving Parsing and PP Attachment Performance with Sense Information. Proceedings of ACL-08: HLT, pages 317-325, Columbus, Ohio, USA (2008)
2. English, J., Nirenburg, S.: Striking a balance: Human and Computer Contributions to Learning through Semantic Analysis. Proceedings of ICSC-2010. Pittsburgh, PA (2010)
3. Ferrucci, D., Brown, E., et al.: Building Watson: An Overview of the DeepQA Project. Association for the Advancement of Artificial Intelligence (2010)
4. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. Proceedings of the

52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60 (2014)
5. McShane, M., Jarrell, B., Fantry, G., Nirenburg, S., Beale, S., Johnson, B.: Revealing the Conceptual Substrate of Biomedical Cognitive Models to the Wider Community. In: Westwood, J.D., Haluck, R.S., et al. (eds.) Medicine Meets Virtual Reality 16, 281 - 286. Amsterdam, Netherlands: IOS Press (2008)
6. McShane, M., Nirenburg, S., Beale, S.: Language Understanding With Ontological Semantics. Advances in Cognitive Systems (forthcoming)
7. McShane, M., Beale, S., Nirenburg, S., Jarrell, B., Fantry, G.: Inconsistency as Diagnostic Tool in a Society of Intelligent Agents. Artificial Intelligence in Medicine (AIIM), 55(3), 137-48 (2012)
8. McShane, M., Nirenburg, S., Jarrell, B.: Modeling Decision-Making Biases. Biologically-Inspired Cognitive Architectures (BICA) Journal, 3, 39-50 (2013)
9. McShane, M., Nirenburg, S.: Use of Ontology, Lexicon and Fact Repository for Reference Resolution in Ontological Semantics. In Oltramari, A., Vossen, P., Qin, L., Hovy, E. (Eds.), New Trends of Research in Ontologies and Lexical Resources, Theory and Applications of Natural Language Processing, Springer, pp 157-185 (2013)
10. McShane, M., Babkin, P.: Automatic Ellipsis Resolution: Recovering Covert Information from Text. Proceedings of AAAI-15 (2015)
11. McShane, M., Nirenburg, S., Babkin, P.: Sentence Trimming in Service of Verb Phrase Ellipsis Resolution. Proceedings of EAP CogSci 2015 (forthcoming)
12. McShane, M., Nirenburg, S., Beale, S.: The Ontological Semantic Treatment of Multi-Word Expressions. Lingvisticae Investigationes (forthcoming)
13. Nirenburg, S., McShane, M., Beale, S.: A Simulated Physiological/Cognitive "Double Agent". In Beal, J., Bello, P., Cassimatis, N., Coen, M., Winston, P. (eds.) Papers from the AAAI Fall Symposium, Naturally Inspired Cognitive Architectures, Washington, D.C., Nov. 7-9. AAAI technical report FS-08-06, Menlo Park, CA: AAAI Press (2008)
14. Nirenburg, S., Raskin, V.: Ontological Semantics. Cambridge, MA: The MIT Press (2004)
15. Piantadosi, S. T., Tily, H., Gibson, E.: The Communicative Function of Ambiguity in Language. Cognition 122, 280-291 (2012)
16. Schank, R., Riesbeck, C.: Inside Computer Understanding. Hillsdale, NJ: Erlbaum (1981)
17. Wilks, Y., Fass, D.: Preference Semantics: A Family History. Computing and Mathematics with Applications 23(2) (1992)
18. Woods, W.A.: Procedural Semantics as a Theory of Meaning. Research Report No. 4627. Cambridge, MA: BBN (1981)