Advances in Cognitive Systems 5 (2017) 25–36 Submitted 12/2016; published 12/2017

Choices for Semantic Analysis in Cognitive Systems

Marjorie McShane MARGEMC34@GMAIL.COM
Cognitive Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180 USA

Abstract

Over the past couple of decades, fundamental semantic analysis has been outside of the purview of mainstream natural language processing, but the recent surge of interest in cognitive computing has returned the notion to center stage. However, what exactly *is* fundamental semantic analysis? What methods can be used to achieve it? What counts as a useful result? How does one measure progress? All of these issues, and many more, define the choice space for building semantic analysis systems. This paper describes key aspects of this choice space and suggests that published descriptions of systems should explicitly state both the choices made and the positive and negative consequences of those choices. This will counter the current tendency for system descriptions to tacitly reflect choices understood only by insiders without overtly motivating or justifying them.

1. Introduction

If we look at a given field at a given moment in history, certain beliefs can be so widely held that they are rarely if ever made explicit in the literature. For example, in the 1980s the natural language processing (NLP) community was pursuing the artificial intelligence (AI) challenge of automating human-level proficiency in natural language understanding, placing computational semantics squarely within its purview. Fast forward to the 2010s and one hardly finds traces of that original goal in reported work: the lion's share of interest and resources in NLP have shifted to statistical, knowledge-lean methods applied to large corpora in service of applications that do not require fundamental language understanding. The dominance of this paradigm has become so complete that published contributions assume, without mention or justification, features that are routine to insiders but are in no way uncontroversial or self-evident. For example, it is taken as axiomatic that shallow processing of large corpora is preferred over deep processing of smaller texts; that low ceilings of quality are acceptable as long as the reported system exceeds its predecessors' evaluation scores; and that manually providing prerequisites to systems for both training and evaluation of task-oriented competitions is acceptable even though the resulting systems cannot perform to their evaluated levels on new, raw inputs. These and similar opinions and preferences can, of course, be supported by argumentation; but so can the opposing views. This means that choices should be made explicit, along with their known benefits and drawbacks.

It is a good time to explicate the design space for semantically-oriented NLP because of the recent upswing of interest in computational semantics, fueled by the current popularity of such paradigms as cognitive computing and the application of statistical methods to semantics. Renewed interest does not imply identifying new needs: the needs of automating language understanding have been well understood since the early days of AI. The question is whether, this time, the community will face the considerable challenges imposed by natural language in a realistic way, without either looming pessimism or head-in-the-sand optimism.

One way to encourage developers to identify their choices in building semantic analysis systems is to make explicit the choice space, effectively turning an open-ended question into a multiple-choice one. The goal of this essay is to provide a scaffolding for that type of description. I say "scaffolding" because a full inventory of choices and options for realizing them could run into the hundreds or even thousands of elements, depending on the descriptive granularity; and the quickest way to squelch progress on understanding a problem space is to require that the treatment be complete. In sketching the design space for semantic analysis systems, I take counsel from the experience of organizing our home library: linger too long on the top-level classification – by language, historical period, strictly alphabetical, fiction or non-fiction, his or hers – and you will live with boxes indefinitely. Much saner to make decisions and live with their imperfections.

2. "Semantics" in Non-Cognitive Systems, Briefly

Although the focus here is on semantics for cognitive systems, which involves computing the contextual meaning of language utterances in terms of the agent's world model, we begin with the briefest overview of how the term "semantics" has been used in statistical NLP.

In statistical NLP, the object of analysis is uninterpreted text strings: there is no level of representation beyond actual words. This means that core semantic phenomena such as ontologically-grounded lexical disambiguation, semantic compositionality, ellipsis, implicature, indirect speech acts, and many more are outside of purview. The reason why such exclusions do not halt the works is that developers of such systems select application areas in which useful results can be achieved despite bypassing meaning. For example, both state-of-the-art machine translation and question-answering systems tend to rely on large, redundant corpora. For machine translation, the corpora include parallel texts in the source and target languages – and if no such corpora exist, then the methods either do not apply or perform poorly. Question-answering systems, for their part, exploit the redundancy of large corpora by orienting around linguistically simple formulations of responses rather than those that would require semantic analysis. For example, to answer "Where was John F. Kennedy born?" such a system will seek a relatively direct formulation such as "John F. Kennedy [or *President Kennedy*, or *JFK*] was born in Brookline, Massachusetts". If the only assertion of that fact in a corpus was is "That is where his son, the future president of the United States, was born a year later", such a system will certainly miss it.

If "semantics" is undertaken by statistical systems, it is not defined as computing the full, contextual meaning of inputs but, rather, as leveraging some select, individual feature(s) that can improve a given application. Let us briefly consider just two semantic subtasks that have proven beneficial to statistical NLP systems: semantic role labeling and distributional semantics.

Semantic-Role Labeling. Semantic roles, otherwise known as *case roles*, indicate the main participants in an event, such as the agent, theme, instrument and beneficiary. Although the roles themselves are semantic entities, in a statistical environment they are used to link *uninterpreted* text strings; so the only semantics in this approach is the role label itself. If a semantic-role labeling system is provided with a set of paraphrases, it should be able to establish the same inventory of semantic role assignments for each. For example, given the sentence set *Marcy forced Andrew to lend her his BMW, Andrew was forced by Marcy into lending her his BMW,* and *Andrew lent his BMW to Marcy because she made him*, a semantic role labeler should recognize that there is a *lend* event in which Andrew is the agent, his BMW is the theme, Marcy is the beneficiary, and Marcy caused the event to occur. Semantic role labeling systems are typically trained using supervised machine learning, relying on the corpus annotations provided in such

resources as PropBank and FrameNet. Among the linguistic features that inform semantic role labelers are the verb itself, including its subcategorization frame and selectional constraints, aspects of the syntactic parse tree, the voice (active vs. passive) of the clause, and the elements' linear positions. As Jurafsky and Martin (2009: 670–1) report, semantic role labeling capabilities have improved performance on tasks such as question answering and information extraction.

Distributional Semantics. As Clark (2015) notes in his historical review, distributional semantics operationalizes the intuitions that "a word is characterized by the company it keeps" (Firth, 1957), and "words that occur in similar contexts tend to have similar meanings" (Turney & Pantel, 2010). Distributional models are good at computing similarities between words – e.g., they can establish that *cat* and *dog* are more similar to each other than either of these is to *airplane,* since *cat* and *dog* frequently co-occur with many of the same words: *fur, run, owner, play,* etc. Moreover, statistical techniques, such as Pointwise Mutual Information, can be used to detect that some words are more indicative of a word's meaning than others: whereas *fur* is characteristic of dogs, frequent words like *the* or *has*, which appear in texts with *dog*, are not. Although distributional semantics has proven useful for such applications as document retrieval, it is not a comprehensive approach to computing meaning since it only considers the co-occurrence of words. Among things it does not consider are the ordering of the words, which can have profound semantic implications (*X attacked Y* vs. *Y attacked X*), their compositionality, which is the extent to which the meaning of a group of words can be predicted by the meanings of component words (*The old man kicked the bucket, may he rest in peace* has nothing to do with the act of kicking an open container),[1] or hidden sources of meaning like ellipsis and implicatures.

To sum up, when a statistical NLP system is called "semantic", this typically means that one or more meaning-related features, which can be computed from uninterpreted text strings, are appended to an engine whose main problem-solving power lies elsewhere. As discussed by Zaenen (2006), it would be unrealistic to expect that the same supervised learning methods that have proven useful for knowledge-lean applications can be applied to computational semantics. The problem is that supervised machine learning requires corpus annotation, and annotating semantic features is orders of magnitude more difficult than annotating syntactic features. Moreover, even if semantic annotation were possible, it is far from clear that the learning methods themselves would work very well over a corpus thus annotated since the annotations will necessarily include meanings not overtly represented by text strings. The answer cannot be that more and more annotations are needed, since that would undermine the statistical community's insistence that its approach to NLP is superior to knowledge-based approaches due to less reliance on manual knowledge acquisition.

Using the term "semantic" to describe primarily non-semantic systems that use just one or two semantic features is unfortunate, in the same way as it is unfortunate that the internet, enhanced by select metadata tags, is called "The Semantic Web". The problem with this use of the term "semantics" is that the meaning becomes diluted, much the same way as the term *awesome* can be used in modern-day American English to describe a tasty cup of coffee. Although semantic shift is natural in languages, this naming practice means that systems that actually attempt human-level meaning analysis must then be renamed accordingly, as treating *real, full, deep, actual, human-level semantics*. Similarly, the internet, when it eventually represents the results of such meaning analysis, will need a new name as well: perhaps *the Super-Semantic Web*?

[1] Work is underway to extend distributional semantics to exploit compositionality (Goyal et al., 2013).

This ends the discussion of how aspects of semantics are being appended to primarily statistical systems. Now we proceed to full semantic analysis and the rich choice space available for developers of cognitive systems who are seeking to operationalize it.

3. Semantics for Cognitive Systems

What *is* semantics for cognitive systems? The best way to answer this question is to work backwards from the meaning-related needs of cognitive systems overall:

- The agent must interpret the results of all perceptual inputs – language, simulated vision, sensors, etc. – in terms of its model of the world. That is, whether an agent sees that a chicken crossed the road, is told this fact by a trusted source, or hears the unmistakable pitter-patter of chicken feet on concrete, it must integrate that knowledge into its memory using the same knowledge representation language.
- The agent must use its resident model of the world – including its current plans, goals, mindreading of the interlocutor, and understanding of the context – to guide its interpretation of perceptual inputs, since they are often highly ambiguous.
- The agent must dynamically integrate its interpretations of multiple channels of perception since, for example, visual stimuli can help to disambiguate language input and vice versa.
- The agent must store all of the new content interpreted in this way so that it can use this information for reasoning about action.

Positing this broad definition of semantics does not imply that all work on semantics for cognitive systems must embrace all aspects at once or to the same degree. That would impose an impossibly heavy burden on anyone attempting to solve any of the subproblems. However, since cognitive systems research naturally gravitates toward the big picture, it is best to begin here, prior to exploring the more specific choices presented below in Sections 3.1 to 3.8.

3.1 Deep Semantics: AI-NLP or Computational Formal Semantics?

Work on deep semantics for agent systems can be roughly divided into two paradigms, which we consider in turn: AI-NLP and computational formal semantics.

AI-NLP is natural language processing carried out with the goals of achieving human-level artificial intelligence. It requires cleaning up the messiness of natural language by translating elliptical, ambiguous, often ill-formed utterances into well-formed, ontologically-grounded propositions that convey the full contextual meaning that would be understood by a person.

For the past half century, the machine reasoning community has worked under the assumption that the NLP community would succeed in automating this process; and, in the interim, developers have manually crafted knowledge structures as input to machine reasoners. (For a historical discussion, see Nirenburg and McShane, 2016.) However, nothing about the translation process is easy: the agent must resolve lexical and referential ambiguities, detect and reconstruct elided categories, recover from production errors (e.g., repetitions and false starts), detect indirect speech acts, analyze non-literal language, reason about implicatures, and properly incorporate what it has learned into its ever-evolving knowledge about types and tokens in the world.

Because of this complexity, AI-NLP is best suited to smaller domains for which agents can be supplied knowledge and reasoning capabilities analogous to those used by people. But this has raised the issue of the cost of knowledge acquisition, referred to by those preferring statistical techniques as "the knowledge acquisition bottleneck". The problem with this moniker is that it

misrepresents the situation. Agents need knowledge not only to understand language but also to make sense of visual stimuli, a highly expectation-driven process in humans (Nilsson, 2014), to make decisions about action, and other intelligent behaviors. So the "knowledge acquisition bottleneck" argument – which derailed work on computational semantics – only makes sense if one asserts that *all* agents must be able to process *all* texts in *all* domains. This is just as unreasonable as saying that *any* given robot must be able to carry out *any* imaginable task.

Consider the capabilities a robotic kitchen helper would need to interpret the statement, "No, wooden, it's Teflon" as meaning "Don't use the metal utensil you just picked up, use a similar one made of wood because if you use the metal one you might scratch the Teflon". We would expect our robot not only to understand and follow such instructions but also to learn that metal utensils, in principle, are not to be used with Teflon, which should inform its future decision making about utensil selection. It is noteworthy, sociologically speaking, that although specialists and nonspecialists alike applaud robotic systems for succeeding at individual tasks such as grasping and moving objects with particular features, they show a baffling resistance to embracing sophisticated language processing if any domain constraints are imposed – which explains the overwhelming preference for statistical systems.

A difficult theoretical and practical question in AI-NLP is 'Where does semantics end and general reasoning begin?' Consider a line from a recent pop song: *My momma don't like you and she likes everyone.*[2] Formally, this is a contradiction but, of course, it is not: "everyone" here does not mean all humans. Instead, it refers to the set of humans the mother happens to know who are not completely awful. Even *she* does not like the awful ones – the singer's former girlfriend included. Although it goes without saying that the cognitive agent must understand such implicatures, it would be unfair to impose on the AI-NLP community responsibility for chasing them all down before claiming any success. So, although I heartily agree with Jackendoff (2007: 257) that if linguists do not take responsibility for the interactions between language, knowledge of the world, and reasoning, then nobody else will, we must offer AI-NLP practitioners milestones for success prior to achieving across-the-board modeling of human cognition.

Early examples of AI-NLP, undertaken before the field's shift toward statistical methods, are Conceptual Dependency theory (Schank, 1972) and Preference Semantics (Wilks & Fass, 1992). Current approaches, whose utility is being validated through incorporation into comprehensive cognitive systems, include the theory of Ontological Semantics, as implemented in OntoAgents (Nirenburg & Raskin, 2004; McShane & Nirenburg, 2012), and the LUCIA language understanding system used by the robot ROSIE (Lindes & Laird, 2016).

Computational Formal Semantics. The issues treated in computational formal semantics largely parallel those of non-computational formal semantics: determining the truth conditions of declarative sentences, interpreting non-declarative sentences based on what would make the declarative variant true, and interpreting quantifiers and other elements of the logical vocabulary. Of course, only a small part of language understanding actually involves truth conditions, meaning that computational formal semantics cannot be considered an all-purpose approach to natural language understanding. Rather, it seeks to solve a specific set of reasoning-oriented problems that are related to language insofar as human thought is expressed in language. One of the topics that distinguishes the computational version from its non-computational counterpart is the use of theorem provers to determine the consistency of databases (Blackburn & Bos, 2005).

[2] From "Love Yourself", written by Justin Bieber and Ed Sheeran.

Computational formal semantics develops methods to reason specifically and only about unambiguous propositions (Blackburn & Bos, 2005). However, since unambiguous propositions are rare in natural language, computational systems cannot use natural language as a metalanguage for representing meaning. Instead, other metalanguages must be used, such as first-order logic. Although the translation from natural language into an unambiguous metalanguage would ideally be carried out automatically using AI-NLP, in practice the inputs to formal semantic reasoning engines are typically written by hand, making these systems – at the current stage of development – only partially automatic. This raises two important issues that divide practitioners of NLP: the judgment of the acceptable germination time between obtaining research results and attaining practical utility, and the acceptable inventory of as-yet unfulfilled prerequisites for a system. Formal semanticists who cast their work as computational assume a long germination time and require nontrivial prerequisites to be fulfilled—most notably, a perfect disambiguating translation from natural language to the metalanguage required for reasoning. Still, they are attempting to treat difficult problems that must be handled by advanced intelligent agents, whenever that stage of development is reached. The alternative point of view, as Church (2011) notes, is that NLP is a purely technological pursuit that should concentrate on near-term results of even minimal utility. Both views have their proponents and are being actively pursued.

3.2 Coverage of Linguistic Phenomena

At any given time, there are hundreds of linguistic phenomena that a system could consider inside or outside its purview. For example, a system may or may not understand sentences that include topic fronting (*Chipmunks I like!*), reduced relative clauses (*The flavor I chose is the one I like best*), metonymies (*Tell the baseball bat that we're about to start*), new lexical coinages using productive affixes (*He's so anti-balloon*), or highly elliptical utterances (*Another?*). Moreover, for each such phenomenon, "treatment" could be defined in various ways. We begin with short overviews of some phenomena that have been more widely discussed in the NLP literature.

Word Sense Disambiguation. This involves selecting the correct contextual meaning for each polysemous word and phrase. Since most nontechnical words are, in fact, polysemous, this can be quite challenging. Two of the subchoices involving word sense disambiguation are: (a) whether to use a highly polysemous lexicon that forces the agent to carry out contextual disambiguation or to artificially constrain the agent's lexicon to the senses expected in the application; and (b) whether to assume that the existing lexicon is complete or to require the agent to learn new words and word senses on the fly (cf. learning below).

Semantic Role Labeling. As explained earlier, this establishes the semantic relationships between events and their participants, with the actual inventory of semantic roles (agent, theme, instrument, etc.) being chosen by developers. For agent systems, in contrast to statistical systems, the semantic roles must link *interpreted meanings* rather than *uninterpreted text strings.* Semantic role labeling is thus most naturally undertaken in conjunction with word sense disambiguation.

Reference Resolution. In agent systems, resolving reference involves linking mentions of objects and events to their anchors in the agent's memory. Textual coreference resolution – a topic extensively pursued within statistical NLP – can sometimes offer useful heuristic evidence toward making the necessary link to memory. Since there are many classes of referring expressions, a given agent system can, at any time, cover all or only a subset of them: personal pronouns, reflexive pronouns, proper names, definite descriptions (noun phrases with *the* in English), indefinite descriptions (noun phrases with *a/an* in English), demonstrative pronouns (*this, that*), referential verbs, and elided referring expressions. These categories can be further

broken down semantically or functionally. For example, a system might treat only those instances of the personal pronoun *it* that are referential and that have a nominal textual antecedent, as in *Look at **my new dress**. Do you like **it**?* Such a system would not be able to treat examples in which *it* was either pleonastic (***It** is raining*) or referential with a propositional antecedent (*I asked him to **sharpen all the knives**. Did he finish **it**?*).[3]

Ellipsis Detection and Resolution. Ellipsis is the non-expression of a category that can be understood from the linguistic or real-world context. In each language, some types of ellipsis are canonical and found in all language genres, whereas other types are constrained to informal genres. For example, in English, verb phrase ellipsis is canonical (*Sally got a new bike but Charlie didn't __*), whereas subject ellipsis is used only informally (*Sorry, __ didn't get your message in time*). Automatically detecting ellipsis is just as challenging as resolving it once detected, particularly in informal genres, in which the extensive use of ellipsis can make utterances look like cobbled together fragments of meaning (*Me, yeah, but not now*) rather than semantically complete structures with elements removed.

Nominal Compounds. These are combinations of two or more nouns whose semantic relationship is not lexically indicated. Given a compound, like *dining room window screen,* an agent, like a person, should understand this is a *screen* for a *window* in the *dining room*. Of course, since natural language prepositions like *for* and *in* are ambiguous, the actual meaning needs to be grounded in an unambiguous ontological metalanguage such as: WINDOW-SCREEN (PART-OF-OBJECT WINDOW (LOCATION DINING-ROOM)). The meaning of a compound is predictable to varying degrees: it can be lexically fixed (*attorney_general*), suggested by a typical ontological pattern (FRUIT + TREE indicates a tree that bears the given type of fruit: *apple tree*), or open to various interpretations depending on the real-world context (e.g., *Microsoft lecture* could mean a *lecture <at, about, given by an employee of, for an employee of*, etc.> *Microsoft*). In some cases, representing the full meaning of a nominal compound requires more than a single relation linking the given concepts: for example, *shrimp boat* means 'a boat that is the location of fishing events whose theme is shrimp' (McShane, Beale, & Babkin, 2014). When analyzing nominal compounds, the agent must disambiguate all component nouns and establish their semantic relationships. It is worth noting that when nominal compounding is undertaken by the statistical NLP community, component nouns are not semantically analyzed, leading to a clash of entity types like that with case role labeling: non-semantic entities (text strings) are linked by semantic labels.

Indirect speech acts. These occur when the form of an utterance does not align with its illocutionary force (i.e., intended meaning). For example, a declarative statement can be used as a request (*I'd really appreciate a cup of coffee*) or as a question (*I'd love to know why you came two hours late*). For practical purposes, we can divide indirect speech acts into at least three categories: lexically canonical, ontologically canonical, and noncanonical. The two examples above represent lexically canonical patterns: *I'd really appreciate…; I'd love to know why…* Although these do not *always* represent indirect speech acts, that is one of their regularly available analyses, and adding associated phrasal senses to the lexicon is akin to recording multiple senses of *table* in the lexicon: it provides the options for analysis, which the agent must evaluate contextually. Ontologically canonical indirect speech acts can be detected by relying on ontological scripts and related rule sets: for instance, mentioning a negative state of affairs may actually

[3] The MUC co-reference task definition incorporates complex rules that exclude many eventualities from its purview (Hirschman & Chinchor, 1997).

be asking the interlocutor to offer assistance in alleviating it (*I'm freezing!*). Finally, there are instances of indirect speech acts that are so idiosyncratic that even people can fail to understand what is implied: *"I just saw the mailman drive by.* [No response.] *Aren't you going to go out and get the mail?" "Why should I? You get it if you want it."* A semantic analysis system might be tasked with treating a broad or narrow inventory of indirect speech acts. In addition, it might be asked to only detect the potential for an indirect speech-act interpretation or to make the final assessment of whether that interpretation was contextually appropriate.

A continuation of this list of linguistic phenomena would include multiword expressions, typical and novel metaphors, metonymy, implicatures, sarcasm, irony, humor, and the full inventtory of canonical and noncanonical syntactic structures. In fact, we could unpack each language-processing module (preprocessing, morphology, syntax, semantics, pragmatics), each knowledge base (lexicon, ontology, fact repository), and each rule set (for parsing, reference resolution, etc.) into its own choice space. Of course, neither here nor in published system reports is it realistic to cite a full inventory. However, it *is* realistic to explain to readers what is going on. For example, if a lexicon contains 50 single-sense words and one word with two senses, then it would not be appropriate to call it a robust, polysemous lexicon that validates the agent's capability for lexical disambiguation. It does not matter if a logician would tell us that this statement is formally true; pragmatically speaking – and we cannot disregard pragmatics – that claim would be misleading. The reason that *true* reporting is so important is that overselling led to disillusionment with AI-NLP 25 years ago, and could doom current work if the scientific community does not make its voice louder than that of industry's crack advertising teams.

3.3 Incremental or Sentence-Level Analysis

The analysis of speech or text input can be carried out incrementally (word by word) or at the level of full sentences. Over the history of NLP, most syntactic parsers have worked at the level of full sentences, since this greatly reduces complexity. However, incremental analysis (e.g., Ball, 2011) is psychologically more plausible and permits the agent to begin acting before an utterance is complete. For example, if someone tells a helper robot, "Take the knife and" the robot can start reaching for the knife before hearing what comes next. Similarly, if it fails to understand something in the middle of an utterance, it should be able to interrupt imme-diately, as a person would. However, incrementality comes at the cost of increased ambiguity in processing subsentential fragments. For example, *take* is a polysemous verb whose meaning is specified only when its arguments are known (*Sally took the knife / a break / my pulse*). Before the direct object has been indicated, *take* will have dozens of candidate interpretations; moreover, when multiple ambiguous words contribute to a fragment, the ambiguity increases accordingly.

The way to cut through this ambiguity is through modeling the agent's *expectations*, which must derive from its dynamically changing understanding of its own plans and goals, its interlocutors' plans and goals, and the speech situation. For example, if a nurse approaches a person with a blood pressure cuff saying, "I'm going to take…", that person has a pretty good idea what is coming next. So, too, should an intelligent agent; but the work involved to operationalize this capability is clearly not limited to language understanding.

An interesting question is whether syntactic parsers that were trained on full sentences can offer useful heuristic evidence when run incrementally in service of a semantic analyzer. We are exploring this issue in the OntoAgent system, using CoreNLP (Manning et al., 2014), and we are finding that syntactic evidence thus acquired can be useful if the agent treats it as overridable.

3.4 Channels of Perception

As mentioned earlier, virtual or robotic language-using intelligent agents can be endowed with channels of simulated perception that can provide the nonlinguistic evidence needed to support full understanding in dialog applications. For example, visual input would permit an agent to interpret gestures, facial expressions, and deictic references to elements of the physical environment ("Give me the green one"), haptic input would enable it to interpret references to the shapes, textures and temperatures of objects ("Hand me the coldest one"), and non-linguistic audio perception would permit it to interpret environmental sounds like crashes and gunshots ("If you hear a gunshot, evacuate"). The earliest AI researchers recognized the need for multimodal inputs to language understanding; however, like other problems that are most naturally approached using knowledge-based modeling, this one has been postponed by the community at large. Current efforts to integrate perceptive inputs are manifest by Lindes and Laird's (2016) ROSIE robot; Scheutz et al.'s (2017) one-shot-learning cognitive robot; Nirenburg, McShane, and Beale's (2008) simulated virtual patients; and Nirenburg and Wood's (2017) language-based learning support for a furniture-building robot.

3.5 What and How Much to Understand?

Although one might think that intelligent agents should fully understand everything that every human in their presence says, this is not characteristic of human behavior, given the messy nature of human speech. Like humans, artificial agents should pay attention to what is in their scope of interest and not exert undue effort to decode what might well defy understanding anyway. What an agent pays attention to, and how confident its understanding must be before acting upon it, depend on the application, which can range from low risk, such as keeping an older person company through conversation (Wilks, 2010), to high risk, such as acting on instructions to detonate explosives. For each domain, agents will have an inventory of capabilities and responsibilities, and must extract relevant inputs from a potentially largely irrelevant language stream. For example, a robot mechanic's assistant might hear: "Shoot, this thing is so stuck… Damn… If I could only… only… Ouch! The wrench, give it here, now!" Here the robot should ignore all of the self-talk – and certainly not ask the human for clarification of what *so stuck* means, which would *really* aggravate him – and wait for something to which it can actually respond: a request to give the speaker the wrench. In short, task-related reasoning must be threaded into the language understanding process (Nirenburg & McShane, 2015), but exactly how that is done – in terms of modeling plans, goals, expectations, and confidence – may vary widely.

3.6 Issues of Domain

We have already established that deep language understanding is most feasible in smaller domains for which we can provide the agent with the knowledge bases and reasoning engines needed to extract full contextual meaning. However, in order to debunk the criticism that "It's only a toy system", developers should explicitly describe each system's domain, scale, and extensibility. Among the relevant data points are the number of words, phrases and ontological concepts covered, whether lexical ambiguity is being treated or avoided (as by recording only one sense for each word and phrase), the man-hour expense of acquiring the knowledge bases, how much of the knowledge is applicable to other domains, the strategy for porting to other domains, and whether the agent independently engages in "lifelong learning" of new words and concepts.

In short, although "toy" has acquired a negative connotation in AI, small systems can have plenty to boast about, but it is the job of developers to provide – not readers to guess at – this context.

3.7 Confidence and Explanation

An important and much discussed question involves when, why, and to what extent people will trust agent systems and, if they do, whether that trust will be justified. One way to make agent systems trustworthy is by enabling them to estimate confidence in all aspects of their processing and to explain both their outputs and their confidence estimations to their collaborators.

The least helpful type of confidence judgment is over a corpus treated as a whole, which is commonly used to evaluate statistical NLP systems. For example, an average of 80% accuracy on reference resolution says nothing about the system's confidence in the resolution of a particular referring expression in a particular context. More helpful are confidence judgments at the level of individual sentences or dialog utterances. Still better are confidence judgments supplied with human-readable explanations of the reasons for a particular level of confidence. For example, assume a system is asked to analyze the sentence *A monkey is eating a banana.* Assume further that the system's lexicon contains only one sense for each of the words, and that the constraints in the corresponding lexicon and ontology entries align perfectly for an interpretation that means "An animal of the type monkey is ingesting a fruit of the type banana". The system could readily provide a very high confidence score as well as this trace of reasoning. By contrast, an input like *It made it because of that* presents a much more difficult ambiguity resolution challenge, due to the presence of the vague and multi-functional expressions *it* and *that,* as well as the light verb *make*, which has dozens of productive and idiomatic senses. In this case, the system's interpretation should have lower confidence due to the complexity of the disambiguation process.

The need for explanation by AI systems has not been central to the field's agenda until recently, as seen by a push toward "explainable artificial intelligence". The problem is that people are increasingly being asked to trust systems whose decision making – based on statistical learning – defies human-tractable explanation (Knight, 2017). When decisions thus made carry weighty consequences – invest in this stock, take this drug – people want reasons they can understand. With respect to explanatory power, intelligent agents that emulate the thinking, learning and decision making of humans have a clear advantage over mathematical approaches. Optimizing the granularity, nature, and even presentation of those decisions remains a wide-open area for research but, clearly, it is best explored through actual systems rather than through high-level theorizing about imagined preferences of imagined system users.

3.8 Evaluation

Evaluation of cognitive systems is a difficult problem. Although most developers would agree that the gold standard is to evaluate an agent's performance on its intended task (Jones, Ray, & van Lent, 2012), there are two problems with casting this as a hard requirement. First, many agent projects that pursue truly sophisticated capabilities have a long incubation period during which developers must show progress to keep their programs viable. Second, work on individual agent capabilities that can contribute to many agent systems is also progress even if it is not immediately integrated for evaluation. A productive alternative to requiring formal evaluations of end systems has been adopted by the *Advances in Cognitive Systems* journal and conference series, which encourage authors to formally state claims and the evidence for them – a framework that embraces a much broader scope of contributions.

4. Conclusions

Although the watering down of word meaning is a natural diachronic process of language, it is particularly striking when the newly vacuous term is "semantics" itself. These days, calling a computer system "semantic" says nothing about how it defines or computes meaning. In fact, the term carries meaning only among like-minded developers who often fail to adequately explain the details of their work. This essay has sketched out the design space for semantic analysis in computer systems and has argued that descriptions of "semantic" systems need to define what is meant by a semantic interpretation, which linguistic phenomena are covered, the extent to which the approach is extensible across domains, and so on, following the inventory of parameters detailed in Section 3. Without this grounding, it is impossible for people within the field – not to mention the public at large – to make sense of reports of scientific and technological advances.

Maintaining a healthy research program on the long-term problem of emulating human-level language understanding in computer systems is as challenging tactically as it is scientifically, particularly since the actual state of the art bears little resemblance to the story of cognitive computing being sold by industry. Although serious researchers favor sober, unadorned reporting – not to mention papers reporting negative results – we all know that they are a hard sell. The goal of this paper is to champion the practice of *telling it straight* as the norm for the still young cognitive systems community. To the extent that understanding the choice space for semantic systems can foster a movement in this direction, this essay will have achieved its goal.

Acknowledgements

This research was supported in part by Grant No. N00014-16-1-2118 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the author and do not necessarily reflect the views of the Office of Naval Research. Thanks to Pat Langley for his insightful commentary on an earlier draft of this paper.

References

Ball, J. (2011). A pseudo-deterministic model of human language processing. *Proceedings of the Thirty-third Annual Conference of the Cognitive Science Society* (pp. 495–500). Austin, TX: Cognitive Science Society.

Blackburn, P., & Bos, J. (2005). *Representation and inference for natural language: A first course in computational semantics*. Stanford, CA: CSLI Publications.

Church, K. (2011). A pendulum swung too far. *Linguistic Issues in Language Technology, 6,* 1–27.

Clark, S. (2015). Vector space models of lexical meaning. In S. Lappin & C. Fox (Eds.), *The handbook of contemporary semantic theory* (2nd ed.). Hoboken, NJ: John Wiley & Sons.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930–1955. In J. R. Firth (Ed)., *Studies in linguistic analysis* (pp. 1–32). Oxford: Blackwell. Reprinted in F. R. Palmer (Ed.) (1968). *Selected papers of J. R. Firth 1952–1959*. London: Longman.

Goyal, K., Jauhar, S. K., Li, H., Sachan, M., Srivastava, S., & Hovy, E. (2013). A structured distributional semantic model: Integrating structure with semantics. *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality* (pp. 20–29). Sofia, Bulgaria: Association for Computational Linguistics.

Hirschman, L., & Chinchor, N. (1997). MUC-7 Coreference Task Definition. Version 3. Retrieved from http://www-nlpir.nist.gov/related_projects/muc/proceedings/co_task.html

Jackendoff, R. (2007). A whole lot of challenges for linguistics. *Journal of English Linguistics,* 35, 253–262.

Jones, R. M., Wray, R. E. III., & van Lent, M. (2012). Practical evaluation of integrated cognitive systems. *Advances in Cognitive Systems, 1,* 83–92.

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Knight, W. (2017). The dark secret at the heart of AI. *MIT Technology Review, 120, May,* 54–63.

Lindes, P., & Laird, J. E. (2016). Toward integrating cognitive linguistics and cognitive language processing. *Proceedings of the Fourteenth International Conference on Cognitive Modeling*. University Park, PA: Pennsylvania State University.

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 Fifty-second Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55–60). Stroudsburg, PA: ACL.

McShane, M., & Nirenburg, S. (2012). A knowledge representation language for natural language processing, simulation and reasoning. *International Journal of Semantic Computing, 6,* 3–23.

McShane, M., Beale, S., & Babkin, P. (2014). Nominal compound interpretation by intelligent agents. *Linguistic Issues in Language Technology, 10,* 1–34.

Nilsson, N. J. (2014). *Understanding beliefs*. Cambridge, MA: The MIT Press.

Nirenburg, S., & McShane, M. (2016). Natural language processing. In S. Chipman (Ed.), *The Oxford handbook of cognitive science* (Vol. 1). New York: Oxford University Press.

Nirenburg, S., & McShane, M. (2015). The interplay of language processing, reasoning and decision-making in cognitive computing. *Proceedings of the Twentieth International Conference on Applications of Natural Language to Information Systems* (pp. 167–179). Passau, Germany: Springer International Publishing.

Nirenburg, S., McShane, M., & Beale, S. (2008). A simulated physiological/cognitive "double agent". *Proceedings of the AAAI Fall Symposium on Naturally Inspired Cognitive Architectures*. Arlington, VA: AAAI Press.

Nirenburg, S., & Raskin, V. (2004). *Ontological semantics*. Cambridge, MA: MIT Press.

Nirenburg, S., & Wood, P. (2017). Toward human-style learning in robots. *Proceedings of the AAAI Fall Symposium on Natural Communication for Human-Robot Collaboration*. Arlington, VA: AAAI Press.

Schank, R. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology, 3,* 532–631.

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 International Conference on Autonomous Agents and Multiagent Systems* (pp. 1378–1386). São Paulo, Brazil: IFAAMAS.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37,* 141–188.

Wilks, Y. (Ed.) (2010). *Close engagements with artificial companions: Key social, psychological, ethical and design issues*. Amsterdam: John Benjamins Publishing Company.

Wilks, Y. & Fass, D. (1992). The preference semantics family. *Computing and Mathematics with Applications, 23,* 205–221.

Zaenen, A. (2006). Mark-up barking up the wrong tree. *Computational Linguistics, 32,* 577–580.