Oxford Handbooks Online

Natural Language Processing

Sergei Nirenburg and Marjorie J. McShane

The Oxford Handbook of Cognitive Science
Edited by Susan E. F. Chipman

Print Publication Date: Oct 2017 Subject: Psychology, Cognitive Psychology
Online Publication Date: Aug 2016 DOI: 10.1093/oxfordhb/9780199842193.013.13

Abstract and Keywords

Natural language processing (NLP) is an important component of cognitive science. Not only is modeling human language capacity an independently motivated scientific pursuit, the prospect of endowing intelligent agents with human-level language processing capabilities has tantalized generations. It is curious, therefore, that most recent work in NLP does not address this goal, instead pursuing so-called “knowledge lean” methods along with applications that can succeed without the computation of meaning. However, against this empiricist backdrop, there are programs of work that seek not only to automate human-level language processing but to seamlessly incorporate it into the overall cognitive functioning of intelligent agents. This chapter describes the origins and development of the field of NLP, major research paradigms, research issues, and tasks. Whenever possible, issues relevant to cognitive modeling are emphasized.

Keywords: natural language processing, semantics, pragmatics, cognitive science, cognitive modeling

Introduction

Natural language processing (NLP) by computers is one of the earliest if not the earliest non-numerical application of computing. Its origins can be traced to the post-WWII flourish of interest and research in the machine translation (MT) of natural languages. Over the seven decades of its existence, NLP—along with the theoretical discipline of computational linguistics that it helped to create—has experienced several surges of public interest as well as a few periods of relative obscurity in the public eye. This fluctuation was in a large measure caused by a series of overoptimistic expectations of imminent commercial breakthroughs that were at best partially realized. Recent successes in NLP, such as online translation systems, have become possible because of

the combination of research progress with the public acceptance of the limitations of the state of the art and, importantly, the spectacular advances in the power and capacity of computers.

The reason that research in NLP is subject to strong fluctuations of fashion and competing practical and theoretical approaches is that, unlike other large-scale scientific efforts, such as mapping the human genome, NLP cannot be circumscribed by a single goal, path, purview, or timeframe. Practitioners' goals range from incrementally improving search engines to generating high-quality machine translations to endowing embodied intelligent agents with language skills rivaling those of a human. Paths of development range from manipulating surface-level strings (words, sentences) using statistical methods, to attempting to analyze select aspects of meaning, to generating full-blown semantic and pragmatic interpretations of text and dialog, which are required to support sophisticated reasoning by artificial intelligent agents. The purview of an R&D effort can range from whittling away at a single linguistic problem (e.g., nominal compounds in the medical domain) to developing theories of language-oriented subdisciplines (p. 338) (e.g., phonology or syntax), to building full-scale, computational language understanding and/or generation systems. Finally, the timeframe for projects can range from months (e.g., developing a system for a competition on named entity recognition) to decades and beyond. Practically the only thing that NLP practitioners do agree on is just how difficult it is to develop computer programs that usefully manipulate natural language—a medium that people master with such ease.

For the uninitiated, the complexities of natural language are not self-evident, so let us begin with just a few examples of what makes language hard for a computer to process. One of the key problems—and what separates natural languages from artificial languages such as those used for computer programming—is widespread ambiguity.[1] Ambiguity refers to the possibility of interpreting something in different ways. There are many types of ambiguity in natural language. Here, we present brief illustrations of a few of them.

1. *Morphological ambiguity*. The Swedish word *frukosten* can have five interpretations, depending on how its component morphemes are interpreted (underscores indicate compound boundaries and plus signs indicate inflectional boundaries before the definite ending): *frukost* + *en* "the breakfast"; *frukost_en* "breakfast juniper"; *fru_kost_en* "wife nutrition juniper"; *fru_kost* + *en* "the wife nutrition"; *fru_ko_sten* "wife cow stone" (Karlsson, 1995, p. 28).
2. *Lexical ambiguity*. The sentence *I made her duck* can have at least the following meanings, depending on how one interprets the words individually and in combination: "I forced her to bend down", "I prepared food out of duck meat for her", "I prepared food out of the meat of a duck that was somehow associated with her (it might have belonged to her, been purchased by her, been raised by her, etc.)", "I made a representation of a duck that is somehow associated with her (maybe she owns it, is holding it, etc.)".

3. *Referential ambiguity*. In the sentence, *The soldiers shot at the women and I saw some of them fall*, who fell—soldiers or women?
4. *Syntactic ambiguity*. In the sentence, *Elaine poked the kid with a stick*, did Elaine poke the kid using a stick, or did she poke (by default, with her finger) a kid who was in possession of a stick?

If these examples served as input to an MT system, the system would, in most cases, have to settle on a single interpretation because different interpretations would be translated differently. (The fact that in some cases ambiguities can successfully be carried across languages cannot be relied on in the general case.) While the need to select a single interpretation should be self-evident for the first two examples, it might be less clear for the latter two, so consider some translation scenarios for (3) and (4), respectively. In Hebrew, the third person plural pronoun—which is needed to translate *them*—has different forms for different genders, so either *women* or *soldiers* must be selected as the co-referent. Similarly, Russian translates the instrumental and accompaniment meaning of *with* in different ways, so this ambiguity must be resolved explicitly.

The most commonplace response to the question of how to settle on one or another interpretation is *Just use the context!* This opinion is widely shared among both linguists and nonspecialists. Of course, people seem to use the context effortlessly, but modeling the same capability in a computer is far from straightforward. After all, what *is* the context, and how can it be categorized, detected, and used? At the risk of some overgeneralization, one can state that both the historical and the contemporary scope of NLP research reflects the variety of responses to the question of how to model and use the context of the speech situation using a computer. At one extreme of the range of solutions is the definition of context as a certain number of words (or other text elements) to the right and to the left of the text element whose interpretation is sought. At the other extreme, the context is a combination of a large number of features including the language signal (e.g., a text) itself, the set of current and stored beliefs of the text producer and consumer, parameters of the speech situation, and the like.

Despite the inherent difficulties of processing natural language and the amorphous nature of the field, NLP has shown significant progress over its half-century history and is currently experiencing an upsurge in creative energy. Of particular interest for this volume is that this surge includes attempts to incorporate semantically oriented language processing into multifunctional intelligent agents using human-inspired cognitive modeling as a tool. But before we address that cutting-edge work, let us start with a very brief historical overview that explains the field's ever-developing understanding of the nature of the language problem and the choice space for advancing science and technology. This overview does not seek to be comprehensive or even fully representative (for additional information, see, (p. 339) e.g., Jurafsky & Martin, 2009, or the survey articles in Clark, Fox, & Lappin, 2010; for an alternative, although compatible, view of NLP in cognitive science see Jackendoff, 2012). Instead, it presents an interpretation of

the current state of the field of NLP in its historical tradition and with a special focus on issues relevant to cognitive science and the development of cognitive systems.

History

NLP was born as machine translation (MT), which was developed into a high-profile scientific and technological area already in the late 1940s. (For historical overviews of MT, see, e.g., Hutchins, 1986; Nirenburg, Somers, & Wilks, 2002.) Within a decade of its development, MT had given rise to the theoretical discipline of computational linguistics and soon thereafter to its applied facets beyond MT that would later come to be referred to as NLP. The eponymous archival periodical of the field, *Computational Linguistics*, started its existence in 1954 as *Mechanical Translation* and in 1965–70 was published as *Mechanical Translation and Computational Linguistics*. A perusal of the journal's table of contents for 1954–70 (http://www.mt-archive.info/MechTrans-TOC.htm) reveals a gradual shift from MT-specific to general computational-linguistic topics. The original MT initiative also influenced other fields of study, most importantly theoretical linguistics and artificial intelligence (AI).

From the outset, MT was concerned with building practical systems using whatever method looked most promising. It is telling that the first programmatic statement about MT—Warren Weaver's famous "memorandum" (1949/1955)—already suggests a few potential approaches to MT that can be seen as seeds of future computational-linguistic and NLP paradigms. Then, as now, such suggestions were influenced by the recent scientific and technological advances that captured the scientific spirit of the times. Today, this may be the Semantic Web, "Big Data," "Deep Learning" or similar visions. Weaver, for his part, was inspired by (a) results in early cybernetics, specifically, McCulloch's artificial neurons (McCulloch & Pitts, 1943) and their use in implementing logical reasoning; (b) recent advances in formal logic; and (c) the spectacular successes of cryptography during World War II, which contributed to the development of information theory, on which Weaver collaborated with Shannon (Shannon & Weaver, 1949/1964). Inspiration from cybernetics can be seen as the seed of the connectionist approach to modeling language and cognition. The formal logic of Tarski, Carnap, and others underwent spectacular development and contributed to formal studies of the syntax and semantics of language as well as to the development of NLP systems. Shannon's information theory is the precursor of the currently ascendant statistical, machine learning-oriented approaches to language processing.

It was understood early on in MT research that simplistic, word-for-word translation could not succeed and that understanding and rendering meaning was essential. It was equally understood that people disambiguate language in context. It is not surprising, therefore, that Weaver suggests involving contextual clues in text analysis:

> If one examines the words in a book, one at a time through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of words. “Fast” may mean “rapid”; or it may mean “motionless”; and there is no way of telling which. But, if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then, if N is large enough one can unambiguously decide the meaning.
>
> (Weaver, 1949/1955)

The context-as-text-window method of analyzing text was in sync with the then ascendant linguistic theory—structuralism—that was imbued with ideas of behaviorism and thus eschewed positing and reasoning about unobservables. A similar approach is a cornerstone of the current corpus-based paradigm in computational linguistics, which usually involves analysis not of text meaning *in toto* but either treatment of the meaning of selected textual strings or no treatment of meaning at all, instead orienting exclusively around the distributional and often also syntactic and morphological properties of words.

It was in the spirit of the times, then, that Weaver did not suggest a goal, let alone a method, of determining text meaning in the framework of MT. The formal study of linguistic meaning was pursued at that time, and for some time thereafter, primarily by philosophers and logicians. The prevailing opinion was that the computational processing of meaning was not possible—this was the reason why Norbert Wiener, a pioneer of cybernetics, refused to join the early MT bandwagon and also why Yehoshua Bar Hillel in the conclusion of his 1960 survey of a decade of MT research insisted that fully automatic, high-quality MT (FAHQMT) could not be (p. 340) an immediate objective of the field before much more work on computational semantics had been carried out. Note that neither Wiener nor Bar Hillel believed that FAHQMT (and, by extension, high-quality NLP) could succeed without the treatment of meaning.

In theoretical work, the newly ascendant school of mentalist theoretical linguists—the generative grammarians—chose to concentrate on the nonsemantic subdiscipline of syntax. Philosophers of language and logicians, by contrast, retained their interest in meaning and its formal representation. This was motivated by their interest in studying formal reasoning. It was assumed that such reasoning could only be carried out over formal representations of the meanings of propositions, and the provenance of these propositions was assumed to be natural language expressions. The accent on working with formal representations was inherited by AI researchers working on building artificial reasoners. Formal languages are more pliant material for logical manipulation than are natural languages, so work on automatic inference making, planning, and theorem proving used as input formal representations of meaning rather than natural language texts. Curiously, the practice of reasoning over formal structures—without much concern about how to derive them automatically from text—can be discerned in logic-based AI

research even today, making Bar Hillel's observation of 1970 as relevant now as it was then:

> The evaluation of arguments presented in a natural language should have been one of the major worries … of logic since its beginnings. However, … the actual development of formal logic took a different course. It seems that … the almost general attitude of all formal logicians was to regard such an evaluation process as a two-stage affair. In the first stage, the original language formulation had to be rephrased, without loss, in a normalized idiom, while in the second stage these normalized formulations would be put through the grindstone of the formal logic evaluator…. Without substantial progress in the first stage even the incredible progress made by mathematical logic in our time will not help us much in solving our total problem.
>
> (Bar Hillel, 1970, pp. 202–203)

Between roughly 1970 and 1990, the flagship research direction in computational linguistics was developing syntactic parsers based on ever more sophisticated formal grammar approaches, such as lexical-functional grammar, generalized phrase structure grammar, and head-driven phrase structure grammar. However, although syntax was a readily treatable subtask of language analysis, syntactic parsing did not fulfill the language processing needs of AI systems. Instead, AI needed NLP systems to address the computation of meaning, since it viewed language processing as a prerequisite to the study of meaning. One simplifying factor, however, was that NLP systems needed to process only that subset of meanings that the given AI system could digest. Still, the task was far from simple. A number of efforts were devoted to extracting and representing text meaning, notably, those of Winograd (1972), Schank and his co-authors (Schank & Abelson, 1977), Wilks (1975), and Woods (1975). The automatic extraction of meaning from text concentrated on the resolution of ambiguity. Disambiguation required knowledge of the context—understood to include the textual context, knowledge about the world, and knowledge about the speech situation.

Such knowledge, which needed to be formulated in machine-tractable form, included, nonexhaustively, grammar formalisms specifically developed to support parsing and text generation, actual grammars developed within these formalisms, dictionaries geared toward supporting automatic lexical disambiguation, rule sets for determining nonpropositional (pragmatic and discourse-oriented) meaning, and world models to support the reasoning involved in interpreting propositions. Acquiring knowledge of all these types for anything beyond small domains proved too expensive in the late 1990s. This realization was responsible for quashing the high hopes—and extensive research outlays—for the field of expert systems of the 1980s. The widespread perception that any attempt to overcome "the knowledge bottleneck" would be futile profoundly affected the path of development of AI in general and NLP in particular: it gave rise to the empiricist—so-called *knowledge-lean*—paradigm of research and development in NLP.

The Empiricist Paradigm

The shift from the knowledge-based to the knowledge-lean paradigm gathered momentum in the early 1990s, and today almost all work on NLP is carried out within this paradigm. Knowledge-lean NLP practitioners considered three primary choices: (1) avoiding the need to address the bottleneck either by pursuing components of applications instead of full applications or (p. 341) by selecting methods and applications that do not rely on extensive amounts of knowledge; (2) seeking ways of bypassing the bottleneck by researching methods that rely on direct textual evidence, not stored knowledge; or (3) addressing the bottleneck head-on but concentrating on learning the knowledge automatically from textual resources, with the eventual goal of using it in NLP applications. Jumping ahead two decades, note that in cognitively inspired approaches to multifunctional agent modeling, the automatic learning of knowledge by the agent has also started to be addressed (Forbus et al., 2007; Navigli, Verlardi, & Faralli, 2011; Nirenburg, Oates, & English, 2007; Wong, Liu, & Bennamoun, 2012).

The search for ways of bypassing the knowledge bottleneck led to the introduction of a broad inventory of new—at least, new for NLP—methods, a process that is still vigorously under way. Many of the methods were imported from other fields of study: statistical and probabilistic approaches, connectionist (i.e., network-oriented) approaches, approaches motivated by decision theory, and approaches imported from econometrics and expert systems in AI. It is noteworthy that the empiricist paradigm was, in fact, already suggested and experimented with in the 1950s and 1960s, for example, by King (1956) with respect to MT. However, it became practical only with the spectacular recent advances in computer storage and processing.

In its purest form, empirical NLP relies on a variety of advanced statistical and probabilistic techniques (including neural nets and Bayesian methods) for measuring similarities and distances between textual elements over increasingly large monolingual or multilingual text corpora—with corpora being viewed as repositories of evidence of human language behavior. In corpus-based approaches, all feature values must be obtained from text corpora and, when available, from annotations manually added to such corpora. Within this neobehaviorist paradigm, there is no need to overtly address unobservables such as meaning; the kind of semantics that does not involve conceptual representation languages to encode natural language meanings has become known as distributional semantics. For example, in the latent semantic analysis approach (Landauer & Dumais, 1997) word meaning is understood essentially distributionally as a list of words that frequently appear in texts within N words of the “target” word whose meaning is being described.

The early flagship application-oriented project in the empiricist paradigm was the Candide MT system (Berger et al., 1994), which was hailed for being able to correctly translate more than 50% of unseen French texts into English. It did not matter that Candide’s performance actually lagged behind the results of the old rule-based SYSTRAN

MT system, which was developed in the 1960s and covered a large number of language pairs, not just French and English (Senellart, Dienes, & Váradi, 2001). The attention of the field was attracted not to the actual results but to the perceived promise of the new approach.

In short, the empiricist paradigm is currently not looking toward producing high-confidence, deep analysis of language aimed at supporting sophisticated reasoning by intelligent agents. Work concentrates on the development of methods and algorithms that offer good prospects of success in the near term. Although this method-oriented approach is undoubtedly legitimate, it is still an open question whether and how well it will support the NLP tasks of greatest concern to cognitive science: understanding how people process language and modeling human-level capabilities in AI agents. For example, it is an open issue how far one can progress in modeling human-level NLP capabilities on the basis of sophisticated reasoning by analogy over vast memories of past stimulus–response pairs (e.g., dialog turns). More specifically, it is unclear whether this method can establish values for features that are needed not only for understanding and generating language but also for other cognitive processes, such as goal-based reasoning and planning, that we do not address in this chapter. A lot of such features are unobservable, meaning that they are not overtly present in the input or output text, which poses an obvious problem for the pure empirical paradigm. As a result, a consensus has developed that, for empiricist NLP's continued progress, the inventory and nature of features it can use must be enhanced. Knowledge-based computational as well as descriptive and theoretical linguistics have been researching a broad set of such features. Manning (2004) suggests that providing features for statistical processing algorithms is what linguists contribute to NLP. The next step is to determine how to procure the values of these features automatically.

The need to compute non-surface feature values makes the empiricist approach less pure by allowing for the use of a variety of resources in addition to plain text corpora: on the one hand, traditional static knowledge resources, such as machine-readable dictionaries and thesauri, grammars, and (p. 342) formal models of the world (ontologies); on the other hand, any kind of text annotation, such as textual metadata and the results of the morphological, syntactic, and semantic analysis of texts. This turn toward using stored knowledge resources as a provenance for feature values effectively forced NLP to revisit the issue of overcoming the knowledge bottleneck. This time, however, the high labor costs that rendered the earlier efforts ineffective had to be contained. Guided by this tenet, knowledge resource acquisition work in the new empiricist paradigm has progressed along several paths, which we now compare with resource acquisition in the knowledge-based paradigm.

The Knowledge Bottleneck Revisited

Work toward overcoming the knowledge bottleneck can be conceptualized as four paradigms: adapting existing, human-oriented lexical and ontological resources for use by machines; manually annotating texts in order to support machine learning in the service of NLP; developing an infrastructure—the Semantic Web—that will use crowdsourcing to provide metadata hypothesized to be of use to NLP engines; and building knowledge resources expressly for NLP—which amounts to disagreeing that there is a bottleneck to begin with.

Approach 1: Adapting Existing Resources to NLP

The 1980s and early 1990s showed a surge of interest in automatically extracting NLP-oriented knowledge bases from machine-readable dictionaries as a means of overcoming the knowledge bottleneck. This research was based on two assumptions: (a) that machine-readable dictionaries contain information that is useful for NLP, and (b) that this information would be relatively easy to extract into a machine-oriented knowledge base (Ide & Veronis, 1993). For example, it was expected that an ontological subsumption hierarchy could be extracted using the hypernyms that introduce most dictionary definitions (*a* ***dog*** *is a domesticated carnivorous* ***mammal***) and that other salient properties could be extracted as well (*a* ***dog*** *… typically has* ***a long snout***). Although information in an idealized lexicon might be both useful and easy to extract, actual dictionaries built by people and for people require human levels of language understanding and reasoning to be adequately interpreted. For example: (a) senses are often split too finely for even a person to detect the differences; (b) definitions regularly contain highly ambiguous descriptors; (c) sense discrimination is often left to examples, meaning that the user must infer the generalization illustrated by the example; (d) the hypernym that typically begins a definition can be of any level of specificity (*a dog is an animal/mammal/carnivore/domesticated carnivore*), which confounds the automatic learning of a semantic hierarchy; (e) the choice of what counts as a salient descriptor is variable across entries (dog: a *domesticated carnivorous* mammal … ; turtle: a *slow-moving* reptile …); and (f) circular definitions are common (*a tool is an implement; an implement is a tool*). After more than a decade's work toward automatically adapting machine-readable dictionaries for NLP, the field's overall conclusion was that this line of research had little direct utility: machine-readable dictionaries simply required too much human-level interpretation to be of much use to machines.

However, traditional dictionaries do not exhaust the available human-oriented lexical resources: the resource called WordNet (Miller, 1995) attempts to record not only what a person knows about words and phrases but also how that knowledge might be organized in the human mind, guided by insights from cognitive science. Begun in the 1980s by George Miller at Princeton University's Cognitive Science Laboratory, the WordNet project has developed a lexical database organized as a semantic network of four directed acyclic graphs, one for each of the major parts of speech: noun, verb, adjective, and adverb. Words are grouped into synonym sets called *synsets*, concepts, or nodes. Synsets within a part-of-speech network are connected by a small number of relations: for nouns, the main ones are *subsumption* ("is a") and *meronymy* ("has as part": *hand ~ finger*); for adjectives, *antonymy*; and for verbs, *troponymy* (indication of manner: *whisper ~ talk*). WordNet itself has few relations across parts of speech, although auxiliary projects have pursued aspects of this knowledge gap.

WordNet was adopted by the NLP community for a similar reason as machine-readable dictionaries: it was large and available. Moreover, its hierarchical structure captured additional aspects of lexical and ontological knowledge that had promise for machine

reasoning in NLP. However, WordNet has proved suboptimal for NLP for the same reasons as machine-readable dictionaries—the vast ambiguity arising from polysemy. For example, *heart* has 10 senses in WordNet: two involve a body part (working muscle vs. muscle of dead animal used as food); four involve feelings (the locus of feelings (p. 343) vs. courage vs. an inclination vs. a positive feeling of liking); two involve centrality (physical vs. nonphysical); one indicates a drawing of a heart-shaped figure; and one is a playing card. For human readers, the full definitions, synonyms, and examples make the classification clear, but for machines they introduce additional ambiguity: for example, the synonym for the "locus of feelings" sense is "bosom," which has eight of its own WordNet senses. So, although the lexicographic quality of this manually acquired resource is very high, interpreting the resource without human-level knowledge of English can be overwhelming—a fact that was experimentally validated when WordNet was leveraged for query expansion in knowledge retrieval applications. Query expansion is the reformulation of a search term using synonymous key words or different grammatical constructions, but, as reported in Gonzalo et al. (Gonzalo, Verdejo, Chugur, & Cigarran, 1998), success has been limited because badly targeted expansion (using synonyms of the wrong meaning of a keyword) degrades performance more than no expansion at all. A relevant comparison is the utility of a traditional monolingual thesaurus to native speakers versus its opaqueness to language learners: whereas native speakers use a thesaurus to jog their memory of words whose meanings and usage contexts they already know, language learners require all of those distinguishing semantic and usage nuances to be made explicit.

Various efforts have been launched toward making the content of WordNet better suited for NLP: for example, select components of some definitions have been manually linked to their correct WordNet senses as a method of disambiguation, and some cross-part-of-speech relations have been added, as between nouns and verbs. Much effort has also been devoted to developing multilingual wordnets and bootstrapping wordnets from one language to another. In the context of this flurry of development, what has *not* been very prevalent is a community-wide assessment of whether wordnets in principle are the best target of the NLP community's resource-building efforts. It is possible that the current conceptualization of wordnets is fundamentally too dependent on the cognitive analysis that the originally envisioned human audience could contribute to the task of analyzing natural language text and that this genre of resource is merely serving as a stopgap until resources developed expressly for NLP have reached a critical size and coverage.

Whereas machine-readable dictionaries and wordnets primarily provide knowledge about *language*, a language processing agent equally requires knowledge about the world to support disambiguation and overall reasoning. For an AI agent, text corpora are the most readily available sources of knowledge about the world. We thus end up with a chicken-and-egg problem: world knowledge is needed to support high-quality NLP, and high-quality NLP is needed for the automatic interpretation of text-based world knowledge. Two lines of work have been devoted to making text corpora more readily useful for NLP

and intelligent agents in general: manual corpus annotation and the development of the Semantic Web.

Approach 2: Text Annotation

At first blush, manual annotation of corpora might seem like the most direct way of facilitating the use of world knowledge by NLP engines: in principle, one could manually record a formal, contextually appropriate interpretation of the meaning of every sentence in a corpus, in a format optimized for machine processing. This would represent a gold standard of semantic analysis. (We cannot call such a structure *the* gold standard because it is inevitable that different annotation schemata will use different metalanguages. Moreover, even when a single metalanguage is used, interannotator agreement is usually not stellar and gets progressively worse as the nature of the annotation task gets more complex, with semantic annotation tending to be more complex than syntactic annotation.) Still, if a large amount of text were annotated this way, then, in principle, it could serve as a training corpus for the machine learning of corresponding annotations, thus addressing the knowledge bottleneck.

Corpus annotation has been in great demand over the past two decades because manually annotated corpora are the lifeline of NLP work based on supervised or semi-supervised machine learning (see, e.g., Abeillé, 2003). These corpora are used as gold standard material for training and evaluating the stochastic algorithms underlying a variety of NLP applications, from named entity recognition to sentiment analysis. However, despite the extensive effort and resources expended on corpus annotation, the annotation of *meaning* has not yet been addressed to a degree sufficient for supporting NLP in the framework of cognitive modeling. So, even though annotated corpora represent a gold standard, the question is, what is the gold in the standard? The value of the gold derives from the task definition for the annotation effort, which in turn derives from combined judgments about practicality and utility on the part of developers.

(p. 344) To date, these judgments have led to creating annotated corpora to support such tasks as syntactic parsing, establishing textual co-reference links, detecting proper names, and calculating light-semantic features, such as the case roles of verbs. Widely used annotated corpora of English include the syntax-oriented Penn Treebank (e.g., Marcus, Santorini, & Marcinkiewicz, 1993; Taylor, Marcus, & Santorini, 2003); PropBank (Palmer, Gildea, & Kingsbury, 2005), which adds semantic role labels to the Penn Treebank; the Automatic Content Extraction (ACE; e.g., Doddington et al., 2004) corpus, which annotates semantic relations and events; and corpora containing annotations of pragmatics-oriented phenomena, such as co-reference (e.g., Poesio, 2004), temporal relations (e.g., Pustejovsky et al., 2005), and opinions (e.g., Wiebe, Wilson, & Cardie, 2005).

Decision making about which phenomena to annotate can be more strongly affected by judgments of practicality than utility: for example, (a) the goal of the Interlingual Annotation of Multilingual Text Corpora project (Dorr et al., 2010) was to create an annotation representation methodology and test it out on six languages, with component phenomena restricted to those aspects of syntax and semantics that developers believed could be consistently handled well by the annotators for all languages; (b) when extending the syntactically oriented Penn Treebank into the semantically supplemented PropBank, developers selected semantic features (co-reference and predicate argument structure) based on the feasibility of annotation (Kingsbury & Palmer, 2002); and (c) the scope of reference phenomena covered by the MUC co-reference corpus was narrowly constrained due to the requirements that the annotation guidelines allow annotators both to achieve 95% interannotator agreement and to annotate quickly and therefore cheaply (Hirschman & Chinchor, 1998).

Before passing an opinion about whether annotation efforts have been sufficiently ambitious, readers should pore over the annotation guidelines compiled for any of the past efforts, which grow exponentially as developers try to cover the confounding messiness of real language use. As Sampson (2003) notes in his thoughtful review of the history of annotation efforts, the annotation scheme needed to cover the syntactic phenomena in his corpus ran to 500 pages—which he likens both in content and in length to the independently produced 300+ page guidelines for Penn Treebank II (Bies, Ferguson, Katz, & MacIntyre, 1995).

Since interannotator agreement and cost are among the most important factors in annotation projects, semi-automation—automatically generating annotations to be checked and corrected by people—has been pursued in earnest. Marcus et al. (Marcus, Santorini, & Marcinkiewicz, 1993) report an experiment revealing that semi-automation of a tagging effort covering parts of speech and light syntax in English was about twice as fast, showed about twice as good interannotator agreement, and was much less error-prone than manual tagging. However, even though semi-automation can speed up and improve annotation for some types of tasks, the cost should still not be underestimated. Brants (2000) reports that although a semi-automated tagging effort covering parts of speech and syntax in German took approximately 50 seconds per sentence, with sentences averaging 17.5 tokens, the actual cost—counting in annotator training, the time for two annotators to carry out the task, for their results to be compared and difficult issues to be resolved—added up to 10 minutes per sentence.

The cost of training and the steepness of the training curve for annotation cannot be overstated. Consider just a few of the rules comprising the MUC-7 task definition (Chinchor, 1997) for the annotation of named entities: family names like *the Kennedys* are not to be annotated, nor are diseases, prizes, and the like named after people: *Alzheimer's, the Nobel prize*. Titles like *Mr.* and *President* are not to be annotated as part of the name, but appositives like *Jr.* and *III* (*"the third"*) are. For place names, compound place names like *Moscow, Russia* are to be annotated as separate entities, and adjectival forms of locations are not to be annotated at all: *American companies*. While there is

nothing wrong with these or any comparable decisions about scope and strategy, lists of such rules are simply very hard to remember—and one must bear in mind that the task of tagging named entities, in the big picture of text annotation, is one of the simplest ones.

This leads us to a seldom discussed but, in our opinion, central aspect of corpus annotation: it is an expensive, labor-intensive resource building endeavor. As long as supervised machine learning relies on annotated corpora, it is not avoiding the knowledge bottleneck, it is simply changing the nature of manual knowledge acquisition. This raises the question of whether machine learning is, actually, a realistic alternative to the widely eschewed knowledge-based paradigm. During the early stages of the neobehaviorist revival, it is understandable that the crucial role of training materials was not (p. 345) fully appreciated; but time has passed and the field has witnessed stagnation in work on many problems for lack of larger, better annotated corpora. Of course, unsupervised learning—the cleanest theoretical concept—avoids the need for training data but its results have, so far, not been overwhelmingly encouraging. In short, the needs of supervised learning methods put the task of corpus annotation, and its concomitant expense, front and center. The complexity and extent of the annotation task is fully commensurate with the task of acquiring knowledge for knowledge-based NLP, showing that the knowledge bottleneck problem simply does not go away with a change in processing paradigms.

Approach 3: The Semantic Web

A corpus that has been treated by many research paradigms is the World Wide Web. The desire to make its content more easily processed by machines has led to a vision known as the Semantic Web, attributed to Berners-Lee and colleagues (e.g., Berners-Lee, Hendler, & Lassila, 2001). They write: “The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.” In effect, the goal is to transform the World Wide Web into a richly annotated corpus, but in ways as yet largely unspecified (see Sparck Jones, 2004, for insightful critique).

We refer to the preceding conception of the Semantic Web as a vision rather than a reality because work toward automatically annotating web pages with machine-interpretable meaning has been largely sidelined by the Semantic Web community in favor of creating formalisms and standards for encoding such meaning, should it ever be pursued in an actual research and development agenda. Moreover, even the simpler desiderata of the Semantic Web community, such as the use of consistent metadata tags, are subject to heavy real-world confounders. Metadata—typically assumed to mean manually provided annotations realized by hypertext tags—is vulnerable to inconsistency, errors, laziness, intentional (e.g., competition-driven) falsification, subconscious biases, and bona fide alternative analyses. Standardization of tags is a topic of intense discussion among the developers, but it is not clear that any practical solution to this problem is

imminent. As a result, especially in critical applications, the metadata cannot currently be fully trusted.

While the current R&D paradigm of the Semantic Web community might ultimately serve some intelligent agents—particularly in applications like e-commerce, in which language understanding is not actually needed (cf. Uschold, 2003)—use of the term "*Semantic* Web" to describe the work is unfortunate, since automatically extracting meaning is centrally absent. As Shirky (2003) writes in his entertaining albeit rather biting analysis, "The Semantic Web takes for granted that many important aspects of the world can be specified in an unambiguous and universally agreed-on fashion, then spends a great deal of time talking about the ideal XML formats for those descriptions. This puts the stress on the wrong part of the problem." In sum, from the point of view of NLP, the web is simply another corpus whose most challenging semantic issues are the same as for any corpus: lexical disambiguation, ellipsis, nonliteral language, and so on. When the web is automatically annotated with this type of analysis, the resulting Super-Semantic Web might approach the original vision of Berners-Lee and colleagues. However, to the best of our knowledge, at present, the Semantic Web community does not consider automatic annotation of web content as a core task.

Approach 4: Building Resources Expressly for NLP

We suggested that the main drawback of applying human-oriented lexical resources to NLP tasks was the machine's inability to contextually disambiguate the massively polysemous words of natural language. Accordingly, a core focus of attention in crafting resources expressly for NLP has been to provide the knowledge to support automatic disambiguation, which necessarily includes both syntactic and semantic expectations about heads and their dependents—most notably, verbs and the arguments they select. As Miller rightly states, "Creating a hand-crafted knowledge base is a labor-intensive enterprise that reasonable people undertake only if they feel strongly that it is necessary and cannot be achieved any other way" (Lenat, Miller, & Yokoi, 1995). Quite a few reasonable groups have seen this task as a necessity, taking different paths toward the same goal. By way of illustration, we briefly compare three: VerbNet, FrameNet, and the integrated lexicon and ontology of the OntoSem language processing environment, which is an implementation of the theory of Ontological Semantics (Nirenburg & Raskin, 2004).

VerbNet (Kipper et al., 2006) is a hierarchical lexicon that was inspired by Levin's (1993) inventory of verb classes. The main theoretical (p. 346) hypothesis underlying Levin's work was that the similarity in syntactic behavior among the members of verb classes suggested a certain semantic affinity. Over the course of its development, VerbNet has expanded Levin's inventory to more than 200 verb classes and subclasses, increased the coverage to more than 4,000 verbs, and described each class in terms of argument structure, legal syntactic realizations of the verb and its arguments, a mapping of the

verb to a WordNet syn-struc, and an indication of coarse-grained semantic constraints on the arguments (e.g., *human, organization*).

FrameNet was inspired by Fillmore's theory of *frame semantics* (Fillmore & Baker, 2009), which suggests that the meaning of most words is best described using language-independent semantic frames that indicate a type of event and the types of entities that participate in it: for example, an *Apply_heat* event involves a *Cook, Food*, and a *Heating_instrument*. A language-independent frame thus described can be *evoked* by given lexical items in a language (e.g., *fry, bake*). The FrameNet resource includes frame descriptions, words that evoke them, and annotated sentences that describe their use. Although FrameNet does include nouns as well as verbs, they are used mostly as dependents in verbal frames.

The OntoSem lexicon describes words and phrases using linked syntactic and semantic structures which, for argument-taking words, include all arguments and certain adjuncts. The ontology describes language-independent concepts (OBJECTs and EVENTs) using a large number of PROPERTIES (e.g., HAS-OBJECT-AS-PART, HAS-EVENT-AS-PART, COST, COLOR, HEIGHT, etc.), including those that link events with the objects that participate in them (i.e., the so-called case-roles, like AGENT, THEME, INSTRUMENT, and so on). The lexicon links to the ontology through its semantic structure zone, which describes the meaning of the head— and, for argument-taking words, the semantic constraints on its arguments—using the ontological metalanguage. Semantic descriptions can include direct mappings to a concept, property-modified mappings to a concept, and even quite elaborate ontologically licensed descriptions that use multiple concepts and properties. For example, the ontology records that the typical AGENT of INGEST is ANIMAL and the typical THEME is INGESTIBLE whose subclasses naturally include BEVERAGE and FOOD. Describing the meaning of the word *ingest* in the lexicon involves simply pointing to the concept INGEST, and indicating the subject of *ingest* (in the active voice) will realize the AGENT case role and the direct object will realize the THEME case role. By contrast, describing the meaning of *eat* or *drink* involves mapping to the same head concept—INGEST—but constraining the THEME to FOOD (for *eat*) and BEVERAGE (for *drink*).

Much could be said about the balance of information recorded in the ontology versus the lexicon: for example, rather than having a single INGEST concept, should the ontology contain INGEST-FOOD and INGEST-BEVERAGE subclasses, thus permitting direct mappings from the words *eat* and *drink*? Interested readers can find some thoughts on this matter in McShane et al. (McShane, Nirenburg, & Beale, 2005). But the main point here is that the OntoSem lexicon and ontology were developed to be used in tandem, with the main goal being to support automatic lexical disambiguation of text. At the time of writing, the OntoSem lexicon contains approximately 27,000 senses covering 12,000 head words, and the OntoSem ontology contains approximately 9,000 concepts.

Comparing FrameNet to the OntoSem lexicon and ontology provides a clear example of how a problem space can be divided differently to align with different theoretical preferences and/or practical ends. Whereas FrameNet combines language-dependent and

language-independent information in a single resource, OntoSem divides these types of knowledge between a lexicon and ontology, which are linked via the lexicon's semantic descriptions. A noteworthy difference between FrameNet and VerbNet, on the one hand, and the OntoSem lexicon/ontology, on the other, is that the OntoSem resources are being developed as one component of the overall OntoSem text processing environment. This does not mean that they can be used only by OntoSem engines, but it does mean that specific knowledge acquisition decisions are affected by the current and planned functionalities of actual processors.

The development of formal ontologies has become a large area in its own right and is not at all always connected with the needs of NLP (see, e.g., Guarino, 1998, for an overview). One of the largest and oldest ontology-building projects to date is Cyc, whose goal is to encode sufficient commonsense knowledge to support any task requiring AI, including but not specifically oriented toward NLP. Doug Lenat, the project leader, described it as a "very long-term, high-risk gamble" (Lenat, 1995) that was intended to stand in contrast to what he called the "bump-on-a-log" projects occupying much of AI (see Stipp, 1995, for a nontechnical (p. 347) perspective). Although initially configured using the frame-like architecture typical of most ontologies, including OntoSem and all ontologies developed using Stanford's open-source Protégé environment (Noy, Fergerson, & Musen, 2000), the knowledge representation strategy quickly shifted to what developers call a "sea of assertions," such that each assertion is equally "about" each of the terms used in it. In a published head-to-head debate with Lenat, Miller, and Yokoi (Lenat et al., 1995), articulates some of the controversial assumptions of the Cyc approach: that commonsense knowledge is propositional, that a large but finite number of factual assertions (supplemented by machine learning of an as-yet undetermined type) can cover all necessary commonsense knowledge, that generative devices are unnecessary, and that a single inventory of commonsense knowledge can be compiled to suit any and all AI applications. Additional points of concern include how people can be expected to manipulate—keep track of, detect lacunae in—a knowledge base containing millions of assertions and the ever-present problem of lexical ambiguity because the assertions are written in unconstrained English. A fair-minded, explanatory review of Cyc in the context of AI can be found in Yuret (1996).

It is not by chance that we conclude our tour of knowledge resources close to where we started: with the issue of lexical ambiguity. As Cyc illustrates, even resources that are specifically compiled to serve machine processing can fail to optimally serve NLP by not directly addressing this problem that so vividly underscores the difference between what humans and machines bring to the table as text processors.

System Components and Integrated Systems

A hallmark of recent NLP has been a widespread preference for developing—often in the context of a competition such as those run by the Message Understanding Conferences (MUC; Grishman & Sundheim, 1996) and Text REtrieval Conferences (TREC; http://trec.nist.gov/)—component technologies over building end-user applications. This preference has usually been justified as "learning to walk before learning to run" or, in a more scholarly fashion, by saying that the scientific method mandates meeting prerequisites for a theory or a model before addressing that theory or model as a whole. In fact, in NLP, the latter precept has been often honored in the breach: in many (most?) cases, theoretical work on specific language phenomena proceeds from the assumption that all the prerequisites for the theory are met, whereas in reality this is seldom the case. This exasperates developers of end-user applications on the lookout for readily available, off-the-shelf components and knowledge resources for boosting the output quality of their applications: their appetites are whetted when they read the description of a theory that promises to help them solve a practical problem only to realize on further investigation that the theory can work only if certain unattainable prerequisites are met. For example, if a theory claims to solve the problem of automatically determining the discourse focus in a dialog but requires a complete propositional semantic analysis of the dialog content as a prerequisite, then it will not be of any use to practical dialog system builders because full semantic analysis is currently beyond the state of the art. It is in this context that one must understand the famous quip by Fred Jelinek, a leader in the field of automatic speech recognition, to the effect that every time he fired a linguist, his system's results improved.

More basic research and development is needed for many component systems to reach human-level quality. For example, even after decades of active development, syntactic parsers of English—to say nothing about other, less researched languages—have not yet attained truly broad coverage or high levels of reliability, particularly for less formal genres, such as dialog. Much thought has been given to understanding and repairing this state of affairs (see, e.g., Clegg & Shepherd, 2007, for an analysis for the biomedical domain). One response has been agreeing to work with results that are known to possibly contain errors. The motivation for this acceptance is practical: progress on certain problems has been slow, and as long as the goal of the field is defined in technological terms—that is, as building application systems that perform specific NLP tasks—one can still hope that imperfect solutions to subproblems and subtasks will suffice.

One reason why imperfect solutions often suffice is because NLP components typically contribute to integrated systems that can buffer the effects of individual errors. In fact, research and development in the area of component integration has become increasingly prominent, with The GATE environment being a case in point (Bontcheva, Tablan, Maynard, & Cunningham, 2004). An additional benefit of the integrative approach is the possibility of viewing components of application systems as black boxes communicating with other components only at the input–output level, thus allowing for the (p. 348) integration of components built using potentially very different approaches. As far as can

be judged from published material, this development paradigm underlies, among others, the modern Internet MT programs (whose output has been widely accepted by users who do not need texts of publication quality) and the *Jeopardy!*-winning Watson answer-questioning system (cf. later discussion).

The Role of System Evaluation

By the time of this writing, the empiricist paradigm in NLP has matured. Its main issues, results, and methods are well presented in the literature (for overviews, see, e.g., Jurafsky & Martin, 2009; Manning & Schütze, 1999). The empiricist paradigm has, in fact, become so dominant that the programs of the major NLP-related conferences of recent years hardly contain contributions from outside this paradigm. A large number of application system prototypes—both component-oriented (e.g., named entity recognition) and end-user ones (e.g., MT)—have been developed and evaluated. Most of the evaluations have been intrinsic, with no reference to the actual utility of the system when it is deployed or, in the case component technologies, without reference to how the component might contribute to the quality of systems into which it is incorporated. A detailed discussion of evaluation in NLP can be found in Resnik and Lin (2010).[2]

Comparing the performance of different systems has proved to be a very expensive undertaking if carried out manually. For example, the expense for the early comparative evaluation of the Pangloss and Candide MT systems (sponsored by the Defense Advanced Research Projects Agency (DARPA)) was on same order of magnitude as the development of the systems themselves. In the new empiricist paradigm in which evaluation occupies the central place, this state of affairs was not acceptable. It is not surprising, therefore, that evaluation of NLP systems emerged as a new area of research. A natural trend in this research is to seek ways to make evaluation less labor-intensive and more automatic. This often results in making evaluation less realistic. Indeed, the utility of some of the evaluation metrics that have come into widespread use has been questioned. For example, in their discussion of the BLEU MT metric, Callison-Burch et al. (Callison-Burch, Osborne, & Koehn, 2006) note that it does not correlate sufficiently well with human judgment because it is geared toward corpus-based systems. They recommend using BLEU only for comparing results from similar corpus-based systems or for evaluating the progress of a single system by evaluating “broad, incremental changes” in its performance over time.

The Knowledge-Based, Cognitive Science-Inspired Paradigm

A quite different avenue of research pursues the scientific goal of modeling human language processing capabilities. Once this goal is adopted, computer systems must strive to match human performance. Modeling human language capabilities requires building integrated systems, not focusing on individual components and phenomena. Integration can be achieved using a variety of computational approaches such as deferred interpretation in pipeline architectures, blackboard architectures, and decision-theoretic approaches. Cognitive architectures such as FORR (Epstein, 1992) accommodate this paradigm for all kinds of cognitive processing, not only NLP. The core prerequisite, though, remains the availability of knowledge to help an intelligent agent system to make decisions while processing language or performing any other reasoning operation.

Knowledge-oriented work is not these days at the center of attention of the NLP community. It is central to the more specialized area of developing reasoning-oriented, multifunctional artificial intelligent agents. NLP systems built in this paradigm are intended for incorporation into broader cognitive systems, pursuing the goal of faithfully replicating human language processing behavior as a part of overall human cognition. The extent, quality, and depth of language processing should be determined by the overall needs of the given cognitive agent, not independently, as when NLP is viewed as an autonomous task. The judgment about how deeply to process a language input must take into account nonlinguistic factors in decision-making such as the long-term and short-term beliefs of the given agent, its biases and goals, and similar features of all other agents in the system's environment.

Consider, for example, anticipatory text understanding, in which an agent can choose to act before achieving a complete analysis of a message and possibly before even waiting for the whole message to come through—being influenced to do so, for example, by the economy of effort principle. Of course, this strategy might lead to errors, but it is undeniable that people routinely pursue anticipatory behavior, making the calibration of the degree of the anticipation an interesting conceptual and technical task for cognitively inspired **(p. 349)** NLP. Anticipatory understanding extends the well-known phenomenon of priming (e.g., Tulving & Schacter, 1990) by involving a broader set of decision parameters than is usually considered, such as the availability of up-to-date values of situation parameters, beliefs about the goals and biases of the speaker/writer, and general, ontological knowledge about the world.

A traditional, although not universally held tenet of knowledge-based NLP has been that, in order to attain human levels of performance, a system needs to extract as much meaning-related knowledge from language inputs as possible. In the framework of cognitive modeling, this supply-side view of NLP should at the very least be juxtaposed with a demand-side view, according to which the needs of the cognitive system as a whole help to determine the scope of work and the temporal ordering of tasks in NLP. In short, the question is, does *this system* need to understand *this bit of input* in order to carry out *its own set of tasks* in accordance with *its currently active goals and plans*? Ideally, systems would be able to make such decisions with the same efficacy as people who

might decide, in a given context, not to ask which particular colleague their spouse is complaining about but, instead, just let him or her let off steam.

It is an open question whether people produce and use comprehensive analyses of the meaning of language inputs in their cognitive functioning. A perfectly plausible hypothesis is that people extract and use only a subset of meaning before acting on it. Moreover, this meaning extraction may be not quite precise in that its results may contain residual ambiguities. This situation reminds one of the debates in the old MT research community about whether to pursue the 100% approach of trying to reach the quality of human translators or be fully satisfied with the 95% approach. Note also that, in the context of cognitive modeling and its applications in robotics and intelligent agent development, it would be neither appropriate nor realistic to require humans and agents to communicate in unambiguous language (Piantadosi, Tily, & Gibson, 2011). So, a corollary of the economy of effort principle suggests that it is appropriate to look for opportunities to avoid processing the entire mass of the ambiguity in the selected inputs, either postponing this process (underspecification) or outright pronouncing it unnecessary (benign ambiguity). This is another potentially very fruitful research direction at the intersection of NLP and cognitive science.

We now turn to a brief survey of some select topics in cognitively inspired NLP.

NLP in Cognitive Architectures

To assess the current views on the role of NLP in computational cognitive science, we turned to a recent authoritative survey of research in cognitive architectures (Langley, Laird, & Rogers, 2009). The survey analyzes nine capabilities that any good cognitive architecture must have: (1) recognition and categorization, (2) decision-making and choice, (3) perception and situation assessment, (4) prediction and monitoring, (5) problem-solving and planning, (6) reasoning and belief maintenance, (7) execution and action, (8) interaction and communication, and (9) remembering, reflection, and learning. The authors primarily subsume NLP under “interaction and communication” but acknowledge that it involves other aspects of cognition as well. The following excerpt summarizes their view. We have added indices in square brackets to link mentioned phenomena with the aspects of cognition just listed:

> A cognitive architecture should … support mechanisms for transforming knowledge into the form and medium through which it will be communicated [8]. The most common form is … language, which follows established conventions for semantics, syntax and pragmatics onto which an agent must map the content it wants to convey… . One can view language generation as a form of planning [5] and execution [7], whereas language understanding involves inference and reasoning [6]. However, the specialized nature of language processing makes these views misleading, since the task raises many additional issues.

(Langley et al., 2009)

Langley et al.’s analysis underscores a noteworthy aspect of most cognitive architectures: even if reasoning is acknowledged as participating in NLP, the architectures are modularized such that core agent reasoning is separate from NLP-oriented reasoning. This perceived dichotomy between general reasoning and reasoning for NLP has been influenced by the knowledge-lean NLP paradigm, which both downplays reasoning as a tool for NLP and uses algorithms that do not mesh well with the kind of reasoning carried out in most cognitive architectures. However, if NLP is pursued within a knowledge-based paradigm, then there is great overlap between the methods and knowledge bases used for all kinds of agent reasoning, as well as the potential for much tighter system integration. Even (p. 350) more importantly, language processing is then, appropriately, not relegated to the input–output periphery of cognitive modeling because reasoning about language is a core task of a comprehensive cognitive model.

Consider, for example, an architecture in which verbal action is considered not separate from other actions (as in Langley et al.’s point 7 vs. point 8) but simply another class of action. Such an organization would capture the fact that, in many cases, the set of plans for attaining an agent’s goal may include a mixture of physical, mental, and verbal actions. For example, if an embodied agent is cold, it can ask someone else to close the window (verbal action), close the window itself (physical action), or focus on something else so as not to notice its coldness (mental action). Conversely, one and the same element of input to reasoning can be generated from sensory, language, or interoceptory (i.e., resulting from the body’s signals, e.g., pain) input or as a result of prior reasoning. For example, a simulated embodied agent can choose to put the goal “have cut not bleed anymore” on its agenda—with an associated plan like “affix a bandage”—because it independently noticed that its finger was bleeding, because someone pointed to its finger and then it noticed it was bleeding (previously, its attention was elsewhere), because someone said “Your finger is bleeding,” or because it felt pain in its finger then looked and saw that it was bleeding.

The conceptual and algorithmic frameworks developed in the fields of agent planning, inference, and reasoning can all be usefully incorporated into the analysis of the semantics and pragmatics of discourse. For example, pioneering work of Cohen, Levesque, and Perrault (e.g., Cohen & Levesque, 1990; Perrault, 1990) demonstrated the utility of approaching NLP tasks in terms of AI-style planning; planning is a first-order concern in the field natural language generation (e.g., Reiter, 2010); and inference and reasoning have been at the center of attention of AI-style NLP for many years.

Returning to Langley et al.’s survey, their section on open issues in cognitive architectures states: “Although natural language processing has been demonstrated within some architectures, few intelligent systems have combined this with the ability to communicate about their own decisions, plans, and other cognitive activities in a general manner.” Indeed, of the 18 representative architectures briefly described in the Appendix, only two—SOAR (Lewis, 1993) and GLAIR (Shapiro & Ismail, 2003)—are

overtly credited with involving NLP, and one, ACT-R, is credited indirectly by reference to applied work on tutoring (Koedinger, Anderson, Hadley, & Mark, 1997) within its framework. Although many cognitive architectures claim to have implemented language processing (13 of the 26 included in a survey by Samsonovich, http://bicasociety.org/cogarch/architectures.htm), most of these implementations are limited in scope and depth, and none of them has language at the center of its scientific interests.

There do exist, however, cognitive architectures that fully integrate language processing with overall agent reasoning. One such is OntoAgent, which is a cognitive architecture that supports the development of language-endowed intelligent agents that collaborate with people in applications (McShane, Beale, Nirenburg, Jarrell, & Fantry, 2012). All OntoAgent agents have a cognitive side, and some have a simulated physical side as well. They are capable of two types of perception: *interoception* and the perception of *linguistic input*. Responses to interoceptive input are remembering a sensation (typically a symptom) and deciding whether or not to do anything about it at the given time. Responses to language input include learning new words, concepts and facts, responding to a question or suggestion, generating a question based on information or advice just provided, and generating advice based on the information provided along with previous knowledge. Agents use agenda-style control and goal- and plan-based simulation. The underlying organization of and knowledge representation for the physiological and cognitive agents are the same. A core architectural aspect of this agent environment is that the interpretations of all types of perception—such as interoception and language understanding—are represented using the same unambiguous, ontologically grounded metalanguage, which is the language in which memories are stored and reasoning is carried out.

In short, OntoAgent takes responsibility for automatically translating messy natural language into an unambiguous metalanguage suited for reasoning following the "translate then reason" approach to NLP discussed earlier in the historical overview. This translation is carried out using the OntoSem approach to language understanding referred to earlier. It relies a specially crafted computational lexicon, a property-rich ontology specially designed to support the core NLP issue of disambiguation, and various types of rule sets to treat individual phenomena, such as reference resolution, multiword expressions, and the interpretation of indirect (p. 351) speech acts. Of course, not all of the automatically generated natural language-to-metalanguage translations are fully correct—remember how ambiguous natural language input can be. However, the broad program of OntoAgent research and development incorporates not only improving semantically oriented knowledge-based NLP, but also teaching agents to detect which aspects of input are most important to understand, teaching agents to evaluate their own confidence in different aspects of language analysis, endowing them with a theory of the minds of other agents in their universe, and integrating their reasoning about language-related issues with the reasoning associated with other perception, decision-making and action modalities.

Computational Formal Semantics

Formal semantics is a venerable area of study in linguistics and the philosophy of language. Here, we will address only a few issues related to some computational aspects of the field. As mentioned earlier, formal computational semantics concentrates on developing methods to reason about unambiguous propositions formulated in a knowledge representation language (for an introductory overview, see, e.g., Blackburn & Bos, 2005). Unambiguous propositions are rare in natural language, so other metalanguages must be used, such as first-order logic. Although the "computational" aspect of "computational formal semantics" would prefer that the propositions be automatically translated from natural language, they most often are either written by hand or manually translated from natural language sources.

The issues treated in computational formal semantics largely parallel those of noncomputational formal semantics: determining the truth conditions of declarative sentences, interpreting nondeclarative sentences based on what would make the declarative variant true, and interpreting quantifiers. Of course, only a small part of language processing actually involves truth conditions, meaning that computational formal semantics cannot be considered a standalone approach to NLP on a level with, say, knowledge-based NLP or empirical NLP. Rather, computational formal semantics seeks to solve a specific set of reasoning-related problems that are related to language insofar as human thought is expressed in language. One of the topics that distinguishes the computational version of formal semantics from its noncomputational counterpart is the use of theorem provers to determine the consistency of databases (Blackburn & Bos, 2005).

Some would argue that it is premature to talk about computational formal semantics: first, little of the work has been implemented (remember, the prerequisite is to translate natural language into the formal language of logic) and, second, some of the hottest issues turn out to be moot when subjected to the simple test of whether the problem actually occurs in natural language. Regarding the latter, in his analysis of the place of formal semantics in NLP, Wilks (2010) reports a thought-provoking finding about a sentence type (*de dicto/de re*) that has been discussed endlessly in the theoretical literature: *John wants to marry a Norwegian*. Such sentences have been claimed to have two interpretations: John wants to marry a particular Norwegian (*de re*), and he wants to marry some Norwegian or other (*de dicto*). When Wilks carried out an informal web search for the corresponding "wants to marry a German," the first 20 hits all had the generic reading, meaning that if one wants to express the specific reading, this turn of phrase just isn't used.

Wilks's overall view of formal computational semantics is "that formal [computational semantic] methods are misdescribed as computation in that they are rarely implemented and that, in the few cases where they are, they are wholly ineffective" (Wilks, 2010). This comment underscores two of the most important features that divide practitioners of

NLP: the interpretation of (1) the acceptable germination time between research results and practical utility and (2) the acceptable inventory of as-yet unfulfilled prerequisites. Formal semanticists who cast their work as computational assume a long germination time and require quite ambitious prerequisites to be fulfilled—most notably, a perfect language-to-metalanguage disambiguating translation. However, they are attempting to treat difficult problems that will arguably need to be handled by advanced intelligent agents, should we ever reach that stage of development. The alternative point of view is that NLP is a practical pursuit that requires near-term results, within which long-term needs tend to be considered less central. Both views, and every combination of them, are supportable and are being actively pursued.

Automatic Processing of Textual Inference

Although at first blush it might seem straightforward to distinguish between what a text means and what inferences it supports, this can actually be quite difficult, as encapsulated by Manning's (2006) paper title, "Local Textual Inference: It's Hard to Circumscribe, but You Know It when You See (p. 352) It—and NLP Needs It." To take just one example from Manning, a person reading *The Mona Lisa hangs in Paris* would be able to infer that *The Mona Lisa is in France*; accordingly, an NLP system with human-like language processing capabilities should be able to make the same inference.

As soon as textual inference was dubbed an NLP task, debate began about its nature, purview, and appropriate evaluation metrics: should systems be provided with exactly the world knowledge they need to make the necessary inferences (e.g., Paris is a city in France), or should they be responsible for acquiring such information themselves? Should language understanding be evaluated separately from reasoning about the world (if that is even possible), or should they be evaluated together, as necessarily interlinked capabilities? Should inferences orient around formal logic ("John has 20 dollars" implies "John has 10 dollars") or naive reasoning ("John has 20 dollars" does not imply "John has 10 dollars"—because he has 20!)? Zaenen et al. (Zaenen, Karttunen, & Crouch, 2005) and Manning (2006) present different points of view on all of these issues, motivated, as always, by different beliefs about the proper scope of NLP, the time frame for development efforts, and all manner of practical and theoretical considerations.

We include textual inference in our short list of phenomena to overview because it underscores the need for NLP to interface with overall agent reasoning.[3] In fact, Manning (2006) promotes the idea of bridging work between the NLP and Knowledge Representation and Reasoning (KR&R) communities, writing: "NLP people can do robust language processing to get things into a form that KR&R people can use, while KR&R people can show the value of using knowledge bases and reasoning that go beyond the shallow bottom-up semantics of most current NLP systems." Note that this is a more bidirectional vision of the interaction between NLP and reasoning than we saw for

computational formal semantics: practitioners of the latter are more likely to expect problems of semantic analysis to be solved independently by the NLP community.

Applications

On the application side, NLP researchers—regardless of their paradigm or paradigms of choice—understandably seek domains and tasks in which lack of breadth of coverage of phenomena (as distinct from corpus coverage) and less-than-perfect quality can be tolerated. Finding a sweet spot for R&D involves matching a set of language processing functionalities, along with the level of confidence with which they can be delivered, to an application that can benefit from those functionalities while tolerating the associated ceiling of confidence. Presented here are short descriptions of some applications that have successfully integrated these features.

- *Web search engines*. Most web search engines do not attempt sophisticated language analysis, nor do they promise high-quality results in language processing or searching overall: the implicit ground rules understood by all Web users is that the search results might contain some documents that might somewhere contain the sought-after information; if a search fails, the user can simply try again. So, although language processing technologies are employed, the real power of search engines lies elsewhere: in their indexing, speed, and search strategies.
- *Web-based translation engines*. Several free translation engines are available on the Internet, offering many combinations of source and target languages. The quality of translations is highly variable, but that is not the point: the point is that people do use these engines, meaning that the difficulty of the MT task is offset by people's willingness to tolerate, and even mentally correct, errors.
- *Information extraction.* Information extraction, undertaken almost exclusively within the empiricist paradigm, has concentrated on named entity extraction, assigning semantic classes to phrases denoting objects, identifying relations among such entities, and identifying instances of events of a given type along with their arguments. As Grishman (2010) explains, "[T]he goal here is to only capture selected types of relations, types of events and other semantic distinctions which are specified in advance." The intent of the enterprise is to make this task tractable—"to identify information which we know how to extract (to some degree of accuracy)" (Grishman, 2010).

- *IBM's Watson*. Watson is the IBM computer system that defeated human opponents on the quiz show Jeopardy! in 2011. Although Watson included many types of state-of-the-art language analysis, it did not assign great weight or responsibility to any single analysis result since a key aspect of the system's problem-solving strategy was to simultaneously apply a large number of algorithms to a vast and largely redundant corpus (Ferrucci et al., 2010). As such, Watson's win (p. 353) cannot be primarily attributed to its language processing (although it certainly helped!), nor can those capabilities be directly ported to applications that require the high-confidence analysis of nonredundant input.
- *Artificial Companions*. Artificial Companions is a relatively new research direction aimed at developing conversational computer programs that can provide people with company, comfort, and pleasant distraction (Wilks, 2010). Early commercial instantiations of a similar idea—such as the furry, rudimentarily language-endowed toy, Furby (by Hasbro)—proved wildly successful, showing people's eagerness to emotionally engage with artificial entities that did not even seek to mimic human sophistication. The Companions application is hypothesized to offer a high-tolerance testbed for the iterative development of NLP and broader AI capabilities.

Considering the historical accent of this overview, one application of particular interest is MT. One of the earliest desiderata in MT was to develop systems that could translate between any language pair, a goal that quite naturally led to research on interlingua. The idea was that natural language input would be translated into an unambiguous metalanguage that would serve as the source for translations into any language. This seemed like it would be an effort-saving approach until attempts to actually implement such systems hit the wall of ambiguity, spawning research on statistical methods that work over parallel corpora. In fact, MT is a star example of a domain for which empirical methods are particularly well suited—at least for languages for which parallel corpora exist.

However, not all languages are prominent enough to offer large parallel corpora with many other languages. As a result, a more recent incarnation of broadly multilingual NLP has been work on so-called low-density languages, defined as having relatively few speakers, little or no descriptive linguistic tradition, and few or no NLP resources. Political and economic considerations have led to an interest in quickly ramping up at least rudimentary translation engines for such languages. While many approaches to this problem constitute primarily engineering efforts, at least two large-scale projects have taken cognitive inspiration from work on "linguistic universals," a long-studied topic in descriptive, theoretical, and field linguistics (Comrie & Smith, 1977). The idea is that all human languages are variations on a cognitively stable theme, which derives from how the human mind processes language. As such, languages can be described by an inventory of parameters that have value sets and means of realization. Describing any given language, therefore, requires understanding which parameters it uses (the set is not truly universal) and how each of the relevant values is realized.

In the Expedition Project, developers compiled an inventory of hundreds of linguistic parameters along with their value sets and known means of realizing those values, and formulated them as a dynamic, open-ended (since previously unattested parameters, values, and realizations were anticipated) decision tree to be filled in by native speakers of any language (McShane & Nirenburg, 2003). For example, the parameter *syntactic function* had values including, non-exhaustively, *subject, direct object* and *indirect object*, and means of realization including word order, affixation and the use of particles; similarly, the parameter *expression of dates* had values such as *month-day-year, month-year*, and means of realization were encoded as patterns of words and numbers. The elicitation of lexicons, inflectional paradigms, productive derivational morphology, and many other features was folded into this parameter-and-values–based strategy. Eliciting knowledge in this structured format permitted the team to develop generic engines to translate any language into English, albeit at a proof-of-concept level of sophistication.

The AVENUE project (Probst & Levin, 2002) gave native speakers of languages other than English a different role. Developers compiled a large inventory of English sentences that were intended to cover a broad range of linguistic universals then asked the native speakers to translate these sentences and manually align the elements of each source–translation pair. Machine learning would then take over to learn transfer rules without ever asking native speakers to explicitly deal with inventories of features or value sets (Carbonell et al., 2002). The rationale behind this approach was to shift the cognitive load from informants to developers: informants were not asked to think in abstract terms about their language, but developers faced the difficult challenges of compiling a sufficient inventory of example sentences and configuring a learner that could draw the necessary conclusions from the resulting bilingual corpus. Both the Expedition and AVENUE projects remain as prototypes which, in our opinion, have very strong potential for success if given further support for development.

(p. 354) We have presented just a few examples of applications of NLP but they underscore the point that, in order to be successful in the real world, applications that incorporate NLP must either use it as just one of many strategies to carry out a larger goal or rely on users to tolerate errors or fill in lacunae.

Conclusion

In the framework of cognitive modeling, NLP has a high potential for development. At present, although NLP is understood as a necessary contributor to cognitive science, it is not at the center of the research activities of most researchers in the field. In applications of cognitive architectures, NLP capabilities are often limited, down to the use of canned phrases as inputs and outputs. The prospect of bringing the NLP capabilities of cognitive agent systems to the level at which other cognitive architecture modules are addressed is exciting and, we would claim, realistic. We hope that NLP for cognitive modeling will

become a core area of research that will continue the NLP work that started within the classical AI period while benefiting from the progress in other NLP paradigms as well as from advances in reasoning and decision-making methodologies.

References

Abeillé, A. (Ed.). (2003). *Treebanks: Building and using parsed corpora*. Dordrecht: Kluwer.

Bar Hillel, Y. (1970). *Aspects of language*. Jerusalem: Magnes.

Berger, A. L., Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Giuett, J. R., Lafferty, J. D., … Urei, L. (1994). The Candide system for machine translation. In *Proceedings of the ARPA Conference on Human Language Technology*, pp. 157–162.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 34–43.

Bies, A., Ferguson, M., Katz, K., & MacIntyre, R. (1995). Bracketing guidelines for Treebank II Style Penn Treebank Project. Available at **http://www.cis.upenn.edu/~bies/manuals/root.pdf**.

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

Bontcheva, K., Tablan, V., Maynard, D., & Cunningham, H. (2004). Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering*. *10*(3–4), 349–373.

Brants, T. (2000). TnT—A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP-2000)*. Seattle, WA, pp. 224–231.

Callison-Burch, C., Osborne, M., & Koehn, P. (2006). Re-evaluating the role of Bleu in machine translation research. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 249–256.

Carbonell, J., Probst, K., Peterson, E., Monson, C., Lavie, A., Brown, R., & Levin, L. (2002). Automatic rule learning for resource-limited MT. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users (AMTA-2002)*, pp. 1–10.

Chinchor, N. (1997). MUC-7 named entity task definition. Version 3.5. September 17, 1997. Available at **http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html**

Clark, A., Fox, C., & Lappin, S. (Eds.). (2010). *The handbook of computational linguistics and natural language processing*. New York, NY: Wiley-Blackwell.

Clegg, A., & Shepherd, A. (2007). Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, *8*(24). Available at **http://www.biomedcentral.com/1471-2105/8/24**.

Cohen, P. R., & Levesque, H. J. (1990). Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, & M. Pollack (Eds.), *Intentions in Communication* (chapter 12, pp. 221–256). Burlington, MA: Morgan Kaufmann.

Comrie, B., & Smith, N. (1977). Lingua descriptive questionnaire. *Lingua*, *42*, 1–72.

Dagan, I., Glicksman, O., & Magnini, B. (2006). The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges: LNAI3944*. SpringerVerlag, pp. 177–190.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., & Weischedel, R. (2004). The automatic content extraction (ACE) program—tasks, data and evaluation. In M. T. Lino, M. F. Xavier, F. Ferreira, R. Costa, & R. Silva (Eds.), *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)* (pp. 837–840).

Dorr, B. J., Passonneau, R. J., Farwell, D., Green, R., Habash, N., Helmreich, S., … Siddharthan, A. (2010). Interlingual annotation of parallel text corpora: A new framework for annotation and evaluation. *Natural Language Engineering*, *16*(3), 197–243.

Epstein, S. L. (1992). Prior knowledge strengthens learning to control search in weak theory domains. *International Journal of Intelligent Systems 7*, 547–586.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., … Welty, C. (2010). Building Watson: An overview of the DeepQA Project. *AI Magazine*, *31*(3), 59–79.

Fillmore, C. J., & Baker, C. F. (2009). A frames approach to semantic analysis. In B. Heine & H. Narrog (Eds.), *The Oxford handbook of linguistic analysis* (pp. 313–340). New York, NY: Oxford University Press.

(p. 355) Forbus, K., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., & Ureel, L. (2007). Integrating natural language, knowledge representation and reasoning, and analogical processing to learn by reading. In *Proceedings of AAAI-07: Twenty-Second AAAI Conference on Artificial Intelligence*, Vancouver, BC, pp. 1542–1547.

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING-ACL Workshop on the Usage of WordNet in Natural Language Processing Systems*, Montreal, pp. 38–44.

Grishman, R. (2010). Information extraction. Chapter 18 in Clark et al., 2010.

Grishman, R., & Sundheim, B. (1996). Message Understanding Conference—6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING), I* (pp. 466–471). Copenhagen.

Guarino, N. (1998). Formal ontology in information systems. In N. Guarino (Ed.), *Formal ontology in information systems* (pp. 3–15). Amsterdam: IOS Press.

Hirschman, L., & Chinchor, N. (1998). MUC-7 coreference task definition. Version 3.0. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

Hutchins, W. J. (1986). *Machine translation: Past, present, future*. Harlow, UK: Longman Higher Education.

Ide, N., & Véronis, J. (1993). Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time? In *Proceedings of KB&KS'93 Workshop* (pp. 257–266). Tokyo.

Jackendoff, R. (2012). Language. In K. Frankish & W. M. Ramsey (Eds.), *The Cambridge handbook of cognitive science* (pp. 171–192). New York, NY: Cambridge University Press.

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

Karlsson, F. (1995). Designing a parser for unrestricted text. In F. Karlsson, A. Voutilainen, J. Heikkilä, & A. Anttila (Eds.), *Constraint grammar* (pp. 1–40). New York, NY: Mouton de Gruyer.

King, G. W. (1956). Stochastic methods of mechanical translation. *Mechanical Translation*, *3*(2): 38–39.

Kingsbury, P., & Palmer, M. (2002). From Treebank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Spain, pp. 1989–1993.

Kipper, K., Korhonen, A., Ryant, N., & Palmer, M. (2006). Extending VerbNet with novel verb classes. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy, pp. 1027–1032.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, *8*, 30–43.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

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

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

Lenat, D., Miller, G., & Yokoi, T. (1995). CYC, WordNet, and EDR: Critiques and responses. *Communications of the ACM*, *38*(11): 45–48.

Levin, B. (1993). *English verb classes and alternations: A preliminary investigation*. Chicago: University of Chicago Press.

Lewis, R. (1993). An architecturally-based theory of human sentence comprehension. PhD Thesis. Carnegie Mellon University. CMU-CS-93-226.

Manning, C. (2004). Language learning: Beyond thunderdome. *Proceedings of Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pp. 138–138.

Manning, C. D. (2006). Local textual inference: It's hard to circumscribe, but you know it when you see it—and NLP needs it. MS, Stanford University. Available at **http://nlp.stanford.edu/~manning/papers/LocalTextualInference.pdf**

Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics 19*(2), 313–330.

McCulloch, W. S., & Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

McShane, M., Beale, S., Nirenburg, S., Jarrell, B., & Fantry, G. (2012). Inconsistency as a diagnostic tool in a society of intelligent agents. *Artificial Intelligence in Medicine*, *55*(3), 137–148.

McShane, M., & Nirenburg, S. (2003). Parameterizing and eliciting text elements across languages. *Machine Translation*, *18*(2), 129–165.

McShane, M., Nirenburg, S., & Beale, S. (2005). An NLP lexicon as a largely language independent resource. *Machine Translation*, *19*(2), 139–173.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Navigli, R., Velardi, P., & Faralli, S. (2011). A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-2011)*. Barcelona, pp. 1872–1877.

Nirenburg, S., Oates, T., & English, J. (2007). Learning by reading by learning to read. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007)*. San Jose, pp. 694–701.

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

Nirenburg, S., Somers, H., & Wilks, Y. (Eds.). (2002). *Readings in machine translation*. ACL-MIT Press Series in Natural Language Processing. Cambridge, Mass.: The MIT Press.

Noy, N. F., Fergerson, R., & Musen, M. A. (2000). The knowledge model of Protégé-2000: Combining interoperability and flexibility. In R. Dieng & O. Corby (Eds.), *Knowledge Engineering and Knowledge Management. Methods, Models and Tools. Proceedings of 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000)*, Juan-les-Pins, France, pp. 17–32.

Palmer, M, Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*(1), 71–105.

Perrault, C. R. (1990). An application of default logic to speech act theory. In P. R. Cohen, J. Morgan, & M. E. Pollack (Eds.), *Intentions in communication* (chapter 9, pp. 161–185). Cambridge, MA: MIT Press.

(p. 356) Piantadosi, S. T., Tily, H., & Gibson, E. (2011). Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, *108*(9), 3526.

Poesio, M. (2004). Discourse annotation and semantic annotation in the GNOME corpus. In B. Webber & D. Byron (Eds.), *Proceedings of the 2004 ACL Workshop on Discourse Annotation*. Barcelona, pp. 72–79.

Probst, K., & Levin, L. (2002). Challenges in automated elicitation of a controlled bilingual corpus. In *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation* (TMI 2002), pp. 157–167.

Pustejovsky, J., Ingria, R., Saurí, R., Castaño, J., Littman, J., Gaizauskas, R. … Mani, I. (2005). The specification language TimeML. In I. Mani, J. Pustejovsky, & R. Gaizauskas (Eds.), *The language of time: A reader* (pp. 545–558). Oxford: Oxford University Press.

Reiter, E. (2010). Natural language generation. In A. Clark, C. Fox, & S. Lappin (Eds.), *Handbook of computational linguistics and natural language processing* (pp. 574–598). New York, NY: Wiley Blackwell.

Resnik, P., & Lin, J. (2010). Evaluation of NLP systems. In A. Clark, C. Fox, & S. Lappin (Eds.), *Handbook of computational linguistics and natural language processing* (pp. 271–296). New York, NY: Wiley Blackwell.

Roemmele, M., Bejan, C., & Gordon, A. (2011). Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proceedings of the International Symposium on Logical Formalizations of Commonsense Reasoning*.

Sampson, G. (2003). Thoughts on two decades of drawing trees. In A. Abeillé (Ed.), *Treebanks: Building and using parsed corpora* (pp. 23–41). Dordrecht: Kluwer.

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

Senellart, J., Dienes, P., & Váradi, T. (2001). New generation SYSTRAN translation system. In *Proceedings of MT Summit VIII: Machine Translation in the Information Age.* Santiago de Compostela, Spain, September 18–22.

Shannon, C. E., & Weaver, W. (1949/1964). *The mathematical theory of communication.* Urbana: University of Illinois Press.

Shapiro, S. C., & Ismail, H. O. (2003). Anchoring in a grounded layered architecture with integrated reasoning. *Robotics and Autonomous Systems*, *43*(2–3), 97–108.

Shirky, C. (2003). The Semantic Web, syllogism, and worldview. First published November 7, 2003, on the “Networks, Economics, and Culture” mailing list. Available at **http://www.shirky.com/writings/semantic_syllogism.html**.

Sparck Jones, K. (2004, December). What’s new about the Semantic Web? Some questions. *ACM SIGIR Form*, *38*(2), 18–23.

Stipp, D. (1995, November 13). 2001 is just around the corner. Where’s Hal?*Fortune*.

Taylor, A., Marcus, M., & Santorini, B. (2003). The Penn Treebank: An overview. In A. Abeillé (Ed.), *Treebanks: Building and using parsed corpora* (pp. 5–22). Dordrecht: Kluwer.

Tulving, E., & Schacter, D. L. (1990). Priming and human memory systems. *Science*, *247*, 301–306.

Uschold, M. (2003). Where are the semantics in the Semantic Web?*AI Magazine 24*(3), 25–36.

Weaver, W. (1949/1955). Translation. Reproduced in W. N. Locke & A. D. Booth (Eds.), *Machine translation of languages: Fourteen essays* (pp. 15–23). Cambridge, MA: MIT Press.

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluations*, *39*(2–3), 165–210.

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

Wilks, Y. A. (1975). Preference semantics. In E. L. Keenan (Ed.), *Formal semantics of natural language: Papers from a colloquium sponsored by the King's College Research Centre* (pp. 321–348). Cambridge: Cambridge University Press.

Winograd, T. (1972). *Understanding Natural Language*. New York, NY: Academic Press.

Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys*, *44* (4), 1–20, 36.

Woods, W. A. (1975). What's in a link: Foundations for semantic networks. In D. G. Bobrow & A. M. Collins (Eds.), *Representation and understanding: Studies in cognitive science* (pp. 35–82). New York, NY: Academic Press.

Yuret, D. (1996, February 13). *The binding roots of symbolic AI: A brief review of the Cyc project.* MIT Artificial Intelligence Laboratory.

Zaenen, A., Karttunen, L., & Crouch, R. S. (2005). Local textual inference: Can it be defined or circumscribed? In *Proceedings of the ACL 2005 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*. pp. 31-36.

Notes:

(1.) There is some ambiguity even in programming languages but it is not nearly as prevalent as in natural languages. For example, consider the polysemy of "nil" in the programming language LISP: it means either "false" or an empty list. Conversely, there is also synonymy, because the empty list can be designated by either () or NIL.

(2.) Resnik and Lin define intrinsic and extrinsic evaluations slightly differently.

(3.) It must be mentioned at this point that at least two directions of work in this area, *recognizing textual entailment* (e.g., Dagan, Glicksman, & Magnini, 2006) and its variant *choice of plausible alternatives* (Roemmele, Bejan, & Gordon, 2011) have engendered NLP competitions and other efforts that predominantly adhere to the currently prevalent distributional, corpus-based methodology that relies on annotating training corpora.

Sergei Nirenburg

Sergei Nirenburg, Cognitive Science Department, Rensselaer Polytechnic Institute

Marjorie J. McShane

Marjorie J. McShane, Cognitive Science Department, Rensselaer Polytechnic Institute