Language Understanding and the Emerging Alignment of Linguistics and Natural Language Processing

James E. Hoard
Research and Technology
Boeing Information and Support Services

1. Overview

The potential impact of natural language processing (NLP) has been widely recognized since the earliest days of computers. Indeed, even as the first electronic computers were becoming a reality, Alan Turing imagined a symbolic processing system--one with true artificial intelligence--that could converse with a person in a way that could not be distinguished from a conversation one might have with a real person. Turing, in his famous article (Turing, 1950), called his thought experiment the Imitation Game. Nowadays, it is called the Turing Test. While no computer program has so far come even remotely close to passing the Turing Test for intelligence, and none will be able to do so at any date in the future that we can reasonably predict, NLP programs that do "understand" language – albeit to a far lesser degree than Turing imagined – will be able to perform many useful and valuable tasks within the next ten-to-twenty years. Among them are these:

1. **Grammar and Style Checking –** Providing editorial critiques of vocabulary usage, grammar, and style – improving the quality of all sorts of writing – especially the readability of complex technical documents.
1. **Machine Translation –** Translating texts, especially business and technical texts, from one natural language to another.
1. **Information Extraction –** Analyzing the meaning of texts in detail, answering specific questions about text content. For many kinds of text (e.g., medical case histories) that are in a well-bounded domain, systems will extract information and put it into databases for statistical analyses.
2. **Natural Language Interfaces –** Understanding natural language commands and taking appropriate actions, providing a much freer interchange between people and computers.
3. **Programming in English –** Enabling the use of carefully controlled, yet ordinary, human language to program computers, largely eliminating much of the need for highly-specialized and arcane computer "languages".
4. **Modeling and Simulation –** Enabling computer modeling and simulation of all manner of real-world activities and scenarios where symbolic information and symbolic reasoning are essential to success.

This informal overview of language understanding and NLP is divided into four sections. Section 2 examines the changing relationship between NLP and linguistics and advances the thesis that the need for language understanding to meet the goals of NLP will have a profound effect on the objectives of linguistics itself and on what qualifies as good linguistic theory and practice. To illustrate this thesis, the third section discusses the scope of language understanding, outlines some fundamental criteria that must be satisfied before any adequate language understanding semantics can be achieved, and offers some suggestions about how one might go about satisfying them. The essential point is that the semantics of

natural language has a logic of its own, which must be dealt with on its own terms, as part of linguistics proper. Section 4 outlines some considerations about approaches to and components for constructing working NLP systems. Section 5 discusses the design and implementation of a grammar and style checker that can determine the senses in which words are being used. (Space limitations preclude taking up any of the other application areas listed above.) The promise of NLP technology is just beginning to be felt in the commercial marketplace. As its commercial impact grows, the effect of NLP on academic linguistics will produce a profound enlargement in its scope and objectives and greatly influence the work of its practitioners. The shift will be, in brief, one that places the present focus on language description, including the concern for language acquisition and linguistic universals, within the much larger (and to my mind, much more interesting) context of language understanding.

2. The Changing Relationship Between Linguistics and Natural Language Processing

Traditionally, work in NLP has been viewed as quite peripheral to linguistics. The relationship was one where NLP received the benefits of linguistic theories and methods, and, at best, imposed perhaps a few requirements on linguistics. Before 1990, commercial and industrial NLP systems were, indeed, few and far between. The entire spectrum consisted of a few machine translation (MT) systems and the odd natural language database query system. Everything else was too primitive or too experimental to have any noticeable impact. The MT systems were non-general and essentially atheoretic, having been built up over a number of years by an accretion of specialized lexicons and procedural code. (See Kay, Gowron, and Norvig, 1994, for an overview of MT systems, of approaches to MT, and for a refreshing discussion of translation as a process of negotiation across languages and cultures.) The database query systems had to be tailored to particular databases and had very limited utility. In this period, NLP did, of course, draw on other disciplines extensively. These areas included computer science, mathematics, and the cognitive sciences. Computational linguistics also played a large role here, since it was the principal source of parsing algorithms and of symbolic processing strategies.

In the mid-1980s, however, a change in the NLP-linguistics relationship started to accelerate. The change came about as NLP practitioners attempted to develop fieldable systems with sufficient coverage to address real-world problems in an acceptable fashion (one which adds value to the users). Constructing robust NLP systems both for grammar and style checking and for information extraction exposed linguistic theories and methods to testing and validation of unprecedented complexity and comprehensiveness. In both areas it quickly became clear that a premier problem is ambiguity resolution (or disambiguation). Systems like these, which are intended to cover a very wide range of inputs, must have comprehensive lexicons and grammars. Yet, the broader the lexical and grammatical coverage, the larger is the potential ambiguity of language analyses produced by the system. That is, in analyzing input text, a robust NLP system must arrive at a preferred interpretation (syntactic, semantic, and/or pragmatic) before any useful action can be taken. The feedback to linguistics was now not just of requirements expressed from a distance, but reports of results (or the lack thereof), and NLP now came to the fore as the arena where linguistic theories and methods are to be tested and validated.

There is now a growing tendency to ensure that linguistic theories are computationally effective. Three examples will serve to illustrate the changing situation: 1) Fraser and Hudson's work on inheritance in word grammar (Fraser and Hudson, 1992)

is indicative of the trend to add computational and NLP support to theoretical work already well underway (Hudson, 1984, 1990). 2) Similarly, Harrison (1988) supplies a full parsing algorithm for generalized phrase structure grammar [GPSG] (Gazdar, Klein, Pullum, and Sag, 1985). 3) In contrast, for head-driven phrase structure grammar [HPSG] (Pollard and Sag, 1987, Pollard and Sag, 1994), now perhaps the most common theoretical framework used in NLP, the development of computational methods was a concern from the beginning. As is the case for HPSG, we can expect in future that linguistic theory and computational algorithms will be developed in tandem and that testing and validation over large-scale problems will be not just "in scope" but, indeed, both customary and mandatory from the outset.

Clearly, testing and validating a (putative) comprehensive set of linguistic rules formulated within some given theory, over a representative domain, is a very difficult task. Success criteria need to be agreed on; and there is no obvious way these can be independently established to everyone's satisfaction. Moreover, different acceptance criteria will be needed depending on the particular language component the rules address and on whether the rules are being evaluated as a stand-alone system or as part of some larger system with which they interact. Evaluating results within a single framework is difficult enough. The evaluation problem is compounded when cross-theory comparisons are attempted. This is a most difficult area, since even agreeing on terminology equivalents that can serve, in part, to bridge theoretical differences, is arduous. (See Neal and Walter, 1991; Harrison, et al, 1991; and Neal, Feit, Funke, and Montgomery, 1992, for some initial contributions to this topic.) The evaluation of comparative system performance is likely to remain both *ad hoc* and not very satisfactory for many years. What can be said now for both within-a-theory comparisons and for cross-theory comparisons is this: While coverage *per se* is the paramount issue (what is correctly analyzed and what is not), so is robustness (the ability of a system to deal with unexpected input), space and time complexity (the resources required for the coverage obtained), extensibility, adaptability, and maintainability.

The trend toward software implementation and large-scale testing and validation in linguistics, driven by NLP application development, ensures that the very objectives of linguistic research will be broadened and deepened. The objectives of linguistic theory before the 1980s were aimed largely at accounting for language structure, not language understanding. The efforts centered on syntax and phonology, with emphasis on language descriptions (synchronic and diachronic), structural universals, language acquisition, and sociolinguistics. Work on semantics, pragmatics, and discourse analysis was a secondary concern. Considering language as a functional system of communication was on the periphery. Given the ambiguity of linguistic expression generally and the fact that people normally interpret verbal and written communication correctly (they "get it right"), this is surprising. It's not that linguists were unaware of ambiguity. Rather, the inclusion of such examples as "time flies like an arrow" and "the shooting of the hunters" in the linguistic literature seemed to be motivated by a requirement to illustrate that one's theory provided a distinct structure (a representation) for each interpretation. That is, the motivation served the needs of descriptive linguistics, and the real issue--how pervasive ambiguity is resolved in everyday language use--was not addressed. And ambiguity is pervasive, arising at all linguistic levels (phonetic, phonological, morphological, lexical, syntactic, semantic, pragmatic), and all of these occur in concert, as it were, in ordinary discourse. The NLP

and computational linguistics literature, in contrast, is chock-full of articles on resolving ambiguity--with numerous approaches and methods proposed for disambiguation, both statistically based and knowledge (rule) based. This situation will not hold. Traditional academic linguistics will indeed need to "get with the program" and broaden its objectives. The change is inevitable and will take place quite quickly, since the people who contribute to linguistics research will be, more often than not in coming years, the very same people who also do work in computational linguistics and NLP.

The development of computational linguistics and the emergence of NLP enables linguists to develop and test theories using large amounts of data. Indeed, it demands and compels them to do so. In brief, linguistics must expand its horizons, augmenting a traditional agenda that is largely limited to descriptive linguistics and representation issues to the much larger--and vastly more difficult--objective of language understanding.

3. Understanding Language

3.1. Meaning, Interpretation, and Speakers' Intentions

The overall goal of natural language processing is to get computers to understand our language, where "our language" is, of course, any language we happen to speak. To anyone who has attempted to design and implement a natural language processing system (or even to anyone who, as a thought experiment, has contemplated doing so), it is obvious that the sheer complexity of language dictates that the goal is at once audaciously difficult and necessarily long-term. No one could hope to get computers to "understand our language" without grounding the enterprise in linguistics, both theory and practice, for that is where the inner workings of language are investigated and described. Many other disciplines have much to contribute to the enterprise. Among them are computer science, mathematics, psychology, philosophy, and cognitive science. Of these, computer science has been the most important, because that is where the methods and limits of computability have been extensively explored and where software engineering methods have been developed. At the intersection of these two disciplines, computational linguistics has flourished and has taken on a vigor of its own. From the 1960s into the 1990s, computational linguistics developed primarily through the work of computer scientists interested in string manipulation, information retrieval, symbolic processing, knowledge representation and reasoning, and natural language processing. Only from the mid-1980s has the linguistic community begun to interact and participate in the development of computational linguistics in a significant way.

The NLP community has been especially interested in analyzing text-based inputs and outputs, primarily because computers readily accept text inputs in standard orthographies, not inputs in a phonetic alphabet (without special provision). Nor, of course, do computers readily accept voice inputs. Using text inputs is also standard practice in linguistics among those who study syntax, semantics, pragmatics, and discourse theory. NLP is complementary to and has much to contribute to the success of speech recognition, speech synthesis, and pen (handwriting) recognition technologies, but, from the NLP point of view, these are extended capabilities.

What do we mean by "understanding" when we talk of language understanding? What would it take to convince us that our computer understands language? It is hard to say precisely, since there is no exact formulation of what we mean by ordinary human understanding of language. The gap between what people do with language in their "native" state – as a matter of course – and what computers can do is profound. In their

"native" state, computers accept strings of ***characters*** as inputs. These character strings have absolutely no meaning or significance to the computer. Any understanding of character strings as natural language is external to the computer and is done at present only by the people who enter the strings, manipulate them (with one sort of application program or another), view them on screen, and print them out.

Now, as a first approximation to language understanding, we would say that a computer understands language if it could represent the meaning of a text (which could be as short as a single sentence) in such a way that it could draw the same conclusions that people do. The kinds of inferences that we would expect our computer to make would include at least the immediate, or shallow, kind. For example, suppose we learn that *Max died on Tuesday*. We can immediately conclude that: *Max died. Max is dead. Max is no longer living. Something happened to Max. Something happened. Someone died. Max used to be alive. Max is not alive now. Max lived up to Tuesday. Max was alive last Monday. There was a death.* – and so forth. Such inferences are shallow in the sense that we draw them immediately from the content of the input text sentence, and we use no information to form our conclusions of the sort that ranges beyond the text we are given. Deeper inferences depend on the extensive knowledge we all have about our culture in general and on any particular knowledge we might have about Max. For instance, we could reasonably conclude, on the basis of cultural expectations, that there will be, in all likelihood, a funeral or memorial service for Max and that the time and place will be announced in the local newspaper. Suppose we also know that Max was the president of the town bank. Then we can conclude that the bank is now without a president, at least temporarily. If we know that Max was married to Abigail, we know that Abigail has been widowed. Given everything people know about the world and what goes on in it, deep reasoning about events and situations that arise can be carried out at will and for as long as one wishes. The number of conclusions we can draw and their significance is open ended.

It seems highly unlikely that one can make a principled distinction between shallow and deep reasoning, claiming that the first is characteristic of and intrinsic to language (and to language understanding) while the second involves general reasoning that goes far beyond language (and far beyond language understanding). Certainly, inferences that apparently follow directly from the meaning of words and their actual use in sentences and discourse seem more basic, even different, from those that follow from broader knowledge of the world. The problem is that it is difficult to see where one sort of reasoning ends and the other begins, for knowledge of the meaning of words is, so far as we know, of the same kind as any other sort of knowledge.

However it is that meaning is represented and that inferences are drawn, for people or for computers, one essential point to keep in mind is that meaning and interpretation are not at all the same thing. In the words of Barwise and Perry (1983:37) "meaning underdetermines interpretation" (See also Barwise and Perry, 1989:61ff). Sperber and Wilson (1988:141) go even further, proclaiming that "the linguistic structure of an utterance grossly underdetermines its interpretation."

Consider the following sentences, which, clearly, do not have the same meaning:

1) Dan turned on the power.
2) Dan threw the switch.

They could easily, however, have the same interpretation, for one possible interpretation for 1)--and also for 2)--is this:

3) Dan pushed upward a lever that is inside the electrical power box on the outside of his house, thereby completing the circuit that supplies his house with electrical power.

Now suppose, however, Dan works for the power company and that the intended interpretation of 1)--and also for 2)--is:

4) Dan reset a large circuit breaker at a substation.

Just as easily, the speaker who uttered 2) could have intended the interpretation to be:

5) Dan physically threw a switch, say, an electrical switch, across the room.

There is clearly a semantic difference in the meaning of "throw" that contributes significantly to the interpretation of 2), in some actual context of use, as 3) or 4), on the one hand, and as 5), on the other. Suppose, though, that the intended interpretation of 2) is 3). Even so, we have underdetermined the situation, since the utterance does not describe the kind of switch nor exactly what Dan did. The actual situation, which we might know through observing Dan, could be this: Dan reached out with his right arm and moved the lever on the main switch upward, using the thumb and index finger of his right hand, thereby completing the electrical circuit and turning on the power to his house. The point is that whatever we take the semantic representation of a sentence to be (or its several semantic representations if it is ambiguous), we have only accounted for its meaning (or its several meanings), not for its actual interpretation in the context in which it is used. In sum, the overt and essentially explicit (or public) meanings of utterances serve as the input to interpretation (a further cognitive endeavor). Any factual correspondence between an interpretation (a mental representation) and the real world (a real-semantic interpretation) is necessarily indirect.

For Barwise and Perry, "Reality consists of situations--individuals having properties and standing in relations at various spatiotemporal locations" (1983:7). That is, situations are states of affairs that are grounded in space and time. Following Pollard and Sag, we will refer to states of affairs as circumstances, where "roughly speaking, circumstances are possible ways the world might be; they are the kinds of things that obtain or do not obtain, depending on how the world is" (1987:86). The circumstance of "Dan's turning on the power" becomes a situation when it is grounded as in utterance 1) at some past time and at some unspecified location. Clearly, to account for situations and circumstances adequately a language understanding system must implement a theory of pragmatics, discourse, and verbal communication. Such a theory must account for a host of phenomena, including those of reference, dialogues, narratives, and discourse relations. The problem of reference is twofold. First, within language, the rules of anaphoric reference must be delineated. Second, the formulation of an adequate theory of reference that holds between language descriptions of things, circumstances, and situations and the actual objects and states of affairs in the real world is also very much at issue. A theory of reference in the second sense stands outside the theory of linguistic semantics, although it very much depends on it.

By and large, the rules of anaphora are not well understood. In particular, the rules for referring to circumstances are not adequately formulated. For example, what is the precise description of *that* in the following pair of sentences?

6) John broke his leg last year.

7) I sure hope that doesn't happen to me.

Evidently, the anaphoric interpretation of *that* requires a procedure that extracts 'break' and its complement structure from the situation, generalizes it to a circumstance, namely, 'X breaking X's leg', and substitutes 'my' for X. The interpretation of *that* is, then, the

circumstance ‘my breaking my leg’ (which is embedded in the circumstance of ‘that not happening to me’, which is embedded in the situation that is the complement of ‘hope’).

While Situation Semantics provides a principled way of describing the information that language communicates--through situations (and situation types), circumstances (and circumstance types), and the relations among situations and circumstances--it does not provide a theory of communication. And, hence, there is no way within the theory of Situation Semantics to constrain the determination of speaker’s intention, which is the goal of language understanding. For a theory of communication we turn to Relevance Theory, as presented by Sperber and Wilson (1986).

Sperber and Wilson’s basic thesis is that a ‘principle of relevance’ governs ‘ostensive-inferential’ communication. The relevance principle is: “Every act of ostensive communication communicates the presumption of its own optimal relevance” (1986:158). Ostensive-inferential communication occurs when: “The communicator produces a stimulus which makes it mutually manifest to communicator and audience that the communicator intends, by means of this stimulus, to make manifest or more manifest to the audience a set of assumptions {I} (1986:155). The presumption of optimal relevance has two parts:

> (a) The set of assumptions {I} which the communicator intends to make manifest to the addressee is relevant enough to make it worth the addressee’s while to process the ostensive stimulus.
>
> (b) The ostensive stimulus is the most relevant one the communicator could have used to communicate {I}” (1986:158).

For Sperber and Wilson, the language understanding task:

> . . .is to construct possible interpretive hypotheses about the contents of {I} and to choose the right one. In different circumstances and different cognitive domains, the task ... may be carried out in different ways. In some cases, it is best carried out by listing all the possible hypotheses, comparing them, and choosing the best one. In others, it is better carried out by searching for an initial hypothesis, testing it to see if it meets some criterion, accepting it and stopping there if it does, and otherwise repeating the process searching for a second hypothesis, and so on. (1986:165).

Kempson (1988:12ff.) briefly discusses some of the similarities and differences between Situation Semantics and Relevance Theory, pointing out that apparent conflicts about the nature of cognitive representations may not be as deep as they seem. Kempson concludes that “the theory of situations does not preclude a system of mental representations” (1988:14). This being so, we are free to use the constructs of Situation Semantics as part of the cognitive language of thought that is at the core of determining speakers’ intentions and of language understanding. As Kempson says: “It is the language of thought that is semantically interpreted, not the natural language expressions. Put crudely, it is our beliefs which are directly about the world we live in, not the sentences of our language” (1988:10). In short, the interpretation task for language understanding requires determining first the meaning of utterances and then the (apparent) intended interpretation of the utterances. Now, it might be the case that the representation of utterance meaning (linguistic semantics expressions) is quite different in kind than the representation of internal cognitive interpretations (propositional semantics expressions). It is a thesis of section 3.2 that this is not so and that a single representation system will suffice for expressing semantic meanings and interpretations.

3.2. Basic Linguistic Elements of Language Understanding Systems

There are a number of basic linguistic elements and capabilities which any model of language understanding must provide, whether we view it as linguistic theory *per se* or as a basis for language understanding systems. To the extent that a given language model fails to satisfy these requirements (in principle or in practice), it is to that extent inherently insufficient for one or another NLP task. It is convenient to separate the capabilities into three categories, one for phonology, morphology, and syntax, one for semantics, and one for pragmatics and discourse. Within all three, there are manifold opportunities for alternative approaches. Since ambiguity is the norm for natural language, and it is the norm at all levels, a fundamental challenge for any language understanding system is to confront ambiguity and resolve it.

Figure 1 shows the conceptual architecture of an information extraction system, suitable for extracting information from (online) texts, that contains a language understanding system as its principal subsystem. (The modules of the Sentence and Discourse Analyzers constitute the language understanding subsystem.) The Preprocessor handles such chores as separating out formatting codes and the like from the basic text stream and segmenting longer texts into pieces appropriate for analysis and information extraction. The Data Extractor contains a set of queries and the rules for applying them. Information extraction is useful whenever there are large numbers of relatively short texts in some domain, the things that go on in that domain share a number of attributes, and it is desirable to "reduce" the goings on to standardized database records. For example, newswire articles on product offerings and sales (in some

industry), financial takeovers and mergers, and stock market trends are suitable domains. Other uses for information extraction include tracking maintenance and repair reports, quality assurance reports, and military tactical messages. (Nicolino (1994) describes a Boeing prototype message processing system which extracts information from military tactical messages, then uses the information to drive a "situation awareness" display. With such a system, a field commander could monitor an entire military operation in near-real time as reports of it arrived and were processed. See also Chinchor, Hirschman, and Lewis (1993) and Chinchor and Sundheim, (1993) for a description and evaluation of a number of message processing systems.)

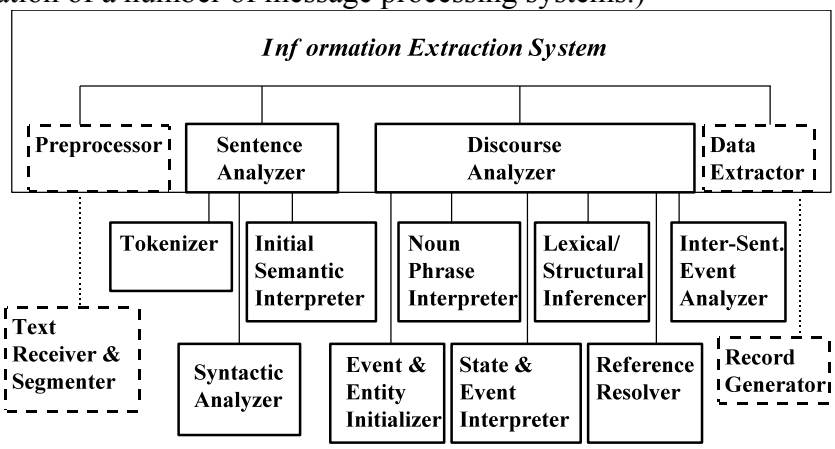


Figure 1. Overall Information Extraction Architecture

3.2.1 Phonology, Morphology, Syntax

Phonology, morphology, and syntax are concerned with the form of language, i.e., with all the tokens of language and with all their permissible concatenations (groupings and arrangements as constituents). Getting a computer to recognize natural language tokens is not as easy as one might suppose, even for a language like English for which the morphology is sparse and all the occurring forms can either be listed or can be easily computed. First, groups of characters and the spaces between them, as they are ordinarily represented in the standard orthography, are only loosely correlated with the morphemes and other lexical units. (The English orthography seems to be typical in this respect). The roots and affixes of inflections, derivations, and compounds must be recognized by some combination of rules and lists. Then, too, multi-word combinations abound. Here are some typical examples: *Las Vegas, Las Cruces, Ingmar Bergman, Ingrid Bergman, De Witt Clinton, Bill Clinton, Vannevar Bush, George Bush, roll back, roll bar, roll call, landing gear, landing strip, bevel gear, pinion gear, slow motion effects, personal effects, liquid assets, liquid crystal display*. Moreover, the list of single-word and multi-word lexical items is unbounded. New proper nouns and multi-word lexical items are constantly added to languages and are regularly encountered in language use. A language understanding system must be prepared to deal with new tokens and combinations of tokens on demand. It is also worth noting that, when punctuation is taken into account, tokenization must be considered on more than one level. For example, one cannot be sure, *a priori* and in isolation, whether *fig.* is an abbreviation for *figure*, whether it represents the word *fig* followed by a sentence-ending period, or whether, indeed, it represents an abbreviation for *figure* at the end of a sentence, where, by convention, only a single period is used. Ambiguity is clearly the norm at the lexical level.

For languages like Arabic, Hebrew, Turkish, and Finnish, which have very complex morphologies, tokenization is a major problem. Some number of (morpho)phonological rules may have applied to any given form., and analyzing (i.e., tokenizing) surface forms can lead to an exceptionally large search space as the rules that might have applied are "undone" to determine the underlying base forms. Because until quite recently the vast majority of the work in computational linguistics has been done on English and similarly "isolating" European languages, computational phonology is an underdeveloped field. It is now receiving much more attention. See especially Koskenniemmi, 1984; Kaplan and Kay, 1994; Maxwell, 1994; Bird, 1995, and Kiraz, 1996.

If one's language understanding system can map character strings into possible tokens, then, given syntactic information about the tokens (part-of-speech and constituent membership possibilities, in particular) we would expect that a comprehensive grammar and a parser could together produce syntactic analyses of sequences of tokens into constituent phrases, clauses, and sentences. (A syntactic parser is an algorithm for applying a grammar to a sequences of tokens. More generically, a parser applies a set of pattern-matching rules to a set of concatenated symbols. A syntactic parser is, then, a particular sort of parser, as is a morphological parser.) When a token sequence is sanctioned by the grammar, a parse, i.e., a description of the sequence, is produced by the parser. There is no guarantee, of course, that the sequences the parser and grammar sanction are actually grammatical. Grammaticality judgments are external to the parser-grammar. Here again the inherent ambiguity of natural language asserts itself. A sentence of twenty words can have dozens, even many dozens, of parses when the analysis is based on syntactic (part-of-speech and constituency) information alone. From this perspective, it does not much matter what syntax formalism one chooses for the grammar. It could be one based on GPSG, HPSG, categorial grammar, tree-adjoining grammar, or some other coherent formalism. The important thing is that the grammar produce (surface) syntactic parses for the actually occurring sequences of tokens. In short, the parser and grammar must together recognize and provide an analysis of the actually occurring sequences of lexical items that speakers of the language agree are acceptable. If the language understanding system is robust, its parser and grammar must handle lexical sequences that, while not completely acceptable, are nonetheless interpretable by speakers of the language. (For example, in the new era of word processing software, writers frequently produce 'sentences' with double articles when rewriting and editing text. Readers are obliged to ignore one of the extra tokens to interpret them successfully.)

3.2.2 Semantics

The semantic interpretation capabilities of any language understanding system depend ultimately on the semantic theory that it implements (however imperfectly). We judge the adequacy of a semantic theory according to at least the following criteria (see Hoard and Kohn, 1994): 1. Partial intentionality, 2. Real-world validity, 3. Multi-valued logic, 4. Inferencing rules, 5. Semantic operators, 6. Coherence conditions, 7. Connectivity, 8. Generalized quantification, 9. Non-arbitrary relation to syntax, 10. Intentionality, 11. Higher-order constructs, and 12. "Amalgamation". Each of these attributes has its functional counterpart in the actual language understanding systems of real language users. They must eventually find functional expression and implementation in one fashion or another in computer-based language understanding systems.

(1) *Partial intentionality*. A language understanding system must achieve its understanding of a verbal or text input in finite time and with finite resources. Real-

time understanding is a highly desirable goal for a computer-based language understanding system. It is, after all, what people are very good at. To meet a real-time objective a language understanding system must provide semantic representations of sentences (actually, of connected discourses) in no worse than linear time as a function of sentence length, and it must do so with a well-bounded amount of memory ("calculation space"). This is not to say that people are computers or use a computer program to understand language. The criterion merely states that any simulation of language understanding using computers must model human capabilities at least to this extent.

(2) *Real-world validity*. Semantic representations must have an overt and explicit character that describes the real world and is consistent with it The representations must fix (or determine) the semantic interpretations and provide one (and only one) possible meaning for any given semantic representation.

(3) Multi-valued logic. To describe the real world of language use, a semantic theory (and a language understanding system) needs at least three truth values, namely, *yes* (true), *no* (false), and *don't know* (indeterminate). These three truth values are required for both open-world and closed-world universe-of-discourse assumptions.

(4) *Inferencing rules*. Being able to draw conclusions that are compatible with the real world and with the knowledge at one's disposal is fundamental to how people use language and to both semantic theory and to language understanding systems. The conclusions one can draw are of at least two different kinds. The first can be called the *means* relationship and is the basis for being able to conclude that *X means Y*. In 2.0, for example, we concluded that *Max died* implies *that Max is no longer living* We did this in part on the basis of the *means* relationship, since, informally, *X dies means that X stops living*. The second relationship can be termed *is covered by* and is the basis for concluding that, say, *Sam built a dory* entails that *Sam built a boat*, for a dory is a kind of boat (i.e. *dory is covered by boat*). Note that we need at least these two kinds of relationships, since we cannot claim either that *dory means boat* or that *boat means dory*.

(5) *Semantic operators*. The operators (or relations) that a semantic theory provides are the basis for deciding how the morphemes in any given sentence are joined to form semantic structures. For instance, in the simple sentence *John loves Mary* we can ask what John's relationship and Mary's relationship is to *loves*. Possible answers are that John is the "cognizer" of *loves,* the one who has a particular cognitive attitude, and that Mary is in the "range" of his cognitive attitude. Neither the exact nature of the semantic operators ('cases', 'valences', and/or 'thematic roles') that a theory may provide nor their number is at issue here. We do, however, postulate a closed set of primitive semantic operators over which semantic structures can be formed. A semantic theory must make substantive claims about how language combines morphemes into semantic structures, admitting some relationships among morphemes and disallowing others, or we cannot construct accounts of situations and circumstances whose real-world validity can, even in principle, be verified.

(6) *Coherence conditions*. To distinguish possible from impossible semantic representations requires, in addition to semantic operators, a set of well-formedness conditions. The set of constraints on combinations of semantic relations provides for

the incoherence of such putative sentences as: *John smiled Mary a watch* (too many complements for *smiled*), *On Wednesday John loved Mary on Tuesday* (two conflicting temporal expressions), *In New York Bob read the book in Boston,* (two conflicting locative expressions), and *Mary knows John swiftly* (manner expression incompatible with a cognitive verb).

(7) *Connectivity*. All the morphemes in a sentence contribute to its meaning and must be accounted for in the semantic representation of the sentence. There are no "sentences" like *John read the book the*, which have "stray" elements (in this example an extra *the*) not integrated into the whole. In those cases when the meaning of a sentence does not result by composition from the meaning of its semantic constituents, we invoke the notion of an idiom to explain the anomaly.

(8) *Generalized quantification*. While the semantics of mathematical proofs can make do with just a universal quantifier and an existential quantifier, human language has an unlimited number of quantifiers. These encompass such variable value quantifiers as *few*, *many,* and *some*, as well as fixed-value quantifiers like *two*, *between three and five*, and *half*. Then, too, natural language quantifiers also include temporal expressions as *frequently, once, occasionally*, and *always*.

(9) *Non-arbitrary relation to syntax*. The relation between syntax and semantics is not arbitrary, but systematic. Any viable semantic theory will have to provide a consistent and effective means to map a syntactic structure to a corresponding semantic representation and from a semantic representation to a corresponding syntactic structure. Because semantic and syntactic structures are of different kinds, there can be no isomorphism.

(10) Relativistic intentionality. Not only are semantic representations but partial descriptions of reality, they are also relative. Total and neutral descriptions of the world using language are impossible, in principle. Any use of language always reflects someone's viewpoint and emphasizes some aspects of a situation to the neglect of others. Different languages use different constructs and devices to describe reality; what is obligatory in the sentences of one language can be absent in another. Furthermore, there is no clear delineation between literal and metaphoric expression. While a semantic theory can provide representations of coherent and meaningful structures, both utterances and the intended interpretations of these utterances, it cannot provide a neutral representation, for the intended interpretation is inevitably a cognitive, mental structure that is determinable only in the context of actual use within a particular language community.

(11) *Higher-order constructs*. Any adequate model of semantics must be able to make higher-order generalizations about language constructs. For instance, verbs that have a cause complement, can have a manner modifier. I.e., if *The fish swam off* is coherent, so is *The fish swam off quickly*.

(12) *Amalgamation*. A semantic theory must be self contained--a notion for which the term 'amalgamation' suggests itself. There cannot be any "meta-language" statements of some sort or other that somehow stand outside the semantic theory and "interpret" it. Statements about semantic theory, if they are meaningful, will necessarily be adequately represented by semantic structures that are expressible within the semantic theory itself. Otherwise, they would themselves need interpretation, and that would require yet another (incomplete) theory of semantics, and so on, endlessly. It follows that pragmatic interpretations and the representation

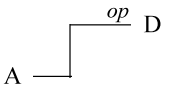
of discourse relations can also be expressed in the same semantic theory used for representing the meaning of individual sentences.

While detailed discussion of these twelve baseline attributes is beyond the scope of this paper, we need to introduce some of the fundamental notions of Cognitive Grammar (Langacker, 1987) and the Relational Logic model of semantics (Hoard and Kohn, 1994) as background to the discussion of a grammar and style checking system.

One of the most salient distinctions in natural language is that between complements and modifiers. The distinction is due to the essential asymmetry among meaningful elements as they are actually used in linguistic constructions. In Langacker's terminology (1987:298ff.), the elements of a sentence are dependent (D) or autonomous (A). The autonomous elements are those that are cognitively (and semantically) complete, requiring no obligatory elaboration by other elements. The autonomous elements are things (and syntactically, nominals). Dependent elements are cognitively incomplete and cannot stand alone in ordinary language use. They require elaboration and appear in construction with at least one other linguistic element. The dependent elements are relationals (and form such syntactic classes as verbs, adverbs, adjectives, articles, conjunctions, and prepositions). Further, a profile is defined as the ***entity*** designated by a semantic structure, and the profile determinant is the component in a construction whose profile is inherited by the composite structure. For example, *girl* is the profile and the profile determinant in *the clever girl*. That is, *girl* is an autonomous element in *the clever girl*, and both *the* and *clever* are dependent elements. The modifier relation is defined as follows: In a construction with autonomous ***element*** A and dependent element D, if A is the profile determinant, then A is the head of the construction, and D modifies (is a modifier of) A. Consider now the prepositional phrase *with the clever girl*. In this case, D is the profile determinant. The complement relation is defined as follows: In a construction with autonomous element A and dependent element D, if D is the profile determinant, the head of the construction is D, and A complements (is a complement of) D. Similarly, there are other relationship types where one dependent element modifies another dependent element (as in *walk fast* and *very tall*), one autonomous element modifies another autonomous element (as in *customer service* and *shake roof*), and one dependent element complements another dependent element (a circumstance complements a situation).

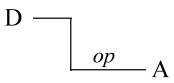
In Relational Logic (RL), semantic structures are represented by directed, acyclic semantic graphs. Relational operators are the edge labels, and morphemes are the nodes. We show the modifier relationship, as in 8a), with a semantic graph that displays D above and to the right of A. The complement relationship is shown, as in 8b), with A displayed below and to the right of D. Diagrams 8c), 8d), and 8e) show schematically relationships where both elements are of the same kind, with one of them being the profile determinant in the construction. As indicated in the semantic graphs, some operator (given here as *op*) always sanctions the relationship between the elements in a construction.

8a)



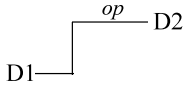
D modifies A

8b)



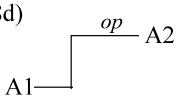
A complements D

8c)



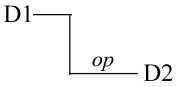
D2 modifies D1

8d)



A2 modifies A1

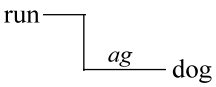
8e)



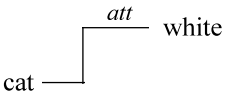
D2 complements D1

Examples of semantic components are given in 9). The operators sanction (in part) the linking of linguistic elements to form components. The operator names suggested in 9), and elsewhere in this paper, are plausible, but in no sense definitive. They are *ag(ent)*, *att(ribute)*, *man(ner)*, *kind*, and *id(em)*. The *idem* operator links situations to circumstances. In 9), it links the situation node, denoted by S, with *love*.

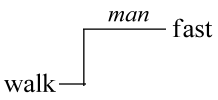
9)



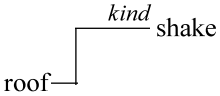
A complements D



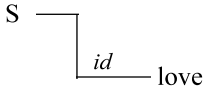
D modifies A



D2 modifies D1



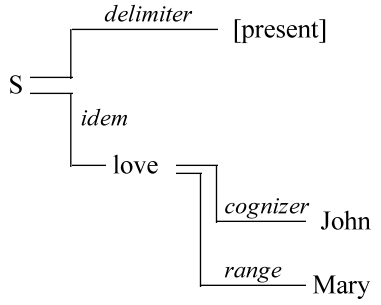
A2 modifies A1



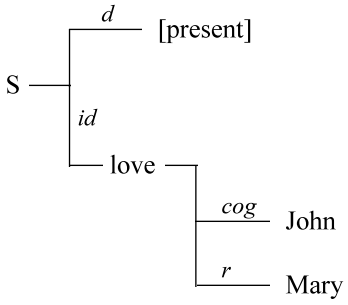
D2 complements D1

The semantic graph for the sentence *John loves Mary* is given in 10a) and 10b). The root of a semantic structure is denoted by the situation node, S (which is indexed in a multi-situation discourse). Here, S is linked to the circumstance "John's loving Mary", which is headed by *love*.

10a)



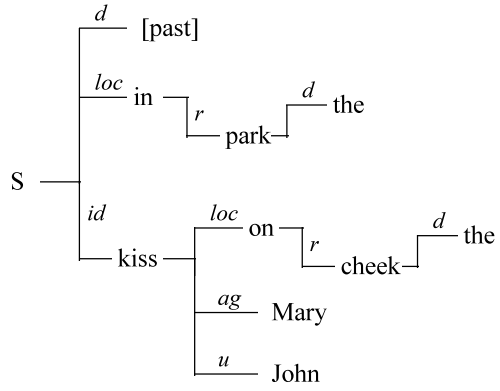
10b)



The two complements of *love*-- *John* and *Mary*--are linked to it by the *cog(nizer)* and *r(ange)* operators, respectively. The present tense morpheme is linked with the *d(elimiter)* operator to S and is displayed above and to the right of the S-node as a situational modifier. For convenience, and without loss of generality, in depicting semantic graphs, we use the abbreviations for the operator labels and superimpose the common portions of independent edges (as shown explicitly in 10a)), drawing them as shown in 10b).

The semantic graph in 11) gives the semantic representation for *Mary kissed John on the cheek in the park*, showing how the semantic representation accounts for the circumstance of "Mary's kissing John on the cheek" (with its internal locative modifier) and situates it in the past and "in the park" (an external modifier). The operators introduced in 11) are *loc(ation)* and *u(ndergoer)*. Note that the delimiter operator sanctions the situating of nominals with determiners (just as it sanctions the situating of verbs with tenses).

11)

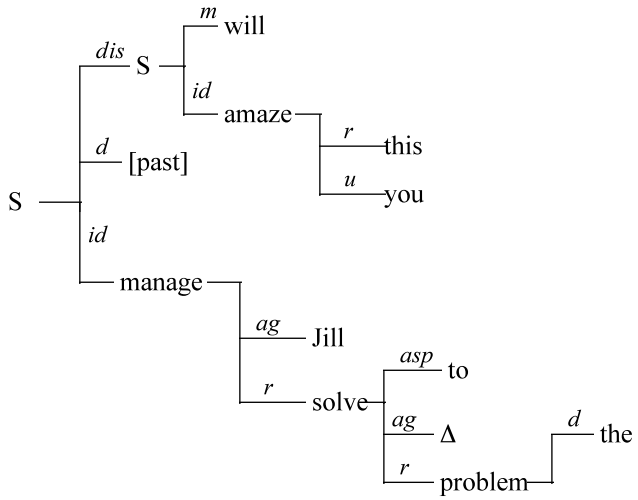


The completeness and closedness of semantic representations in RL guarantees a path between any two morphemes in a sentence. For instance, in example 11) the path from *Mary* to *cheek* goes through *kiss* and *on*; the path between *Mary* and *John* goes just through *kiss*; and the path from *park* to *John* goes through *in*, S, and *kiss*. Note that the explicit (and initial) semantic representation of the sentence does not indicate whose cheek was kissed. It is, in fact, part of the meaning of kiss that the agent does it with the lips, that

the default location for the undergoer is also the lips, and that an explicit internal modifier location is that of the undergoer. In short, we infer that it is John's cheek that was kissed.

Example 12) gives the semantic representation for *Jill--this will amaze you--managed to solve the problem*, a sentence which illustrates both a situation embedded in a situation and a circumstance embedded in a circumstance. (The operators introduced in this example are *dis(junct)*, *m(odal)*, and *asp(ect)*.) The disjunct is a modifier of the matrix sentence, while the 'solving' circumstance is a complement of the 'managing' circumstance. The 'solving' circumstance has an understood agent (indicated with Δ) that we infer is *Jill*.

12)



The incorporation into RL semantic representations of such basic cognitive distinctions as that between dependent and autonomous elements and between modifier and complement is a step toward a linguistic semantics with real-world validity. RL semantic representations require semantic operators--one for every edge--to sanction semantic components, with no morpheme appearing in any semantic graph without explicit sanction, and with no morpheme being left out of account. Hence, on the presumption that every morpheme makes some contribution to the meaning of the sentences in which it appears, RL provides a principled approach to explaining the cognitive basis of semantic coherence and connectedness. Indeed, RL offers some hope of accounting for why a sentence is traditionally described as a 'complete thought' and of illuminating such concepts as 'the language of thought', for we will use RL semantic graphs both for what is explicitly stated and for everything inferred from what is stated. Among the inferences will be, whenever the inferencing is successful, a representation of the speaker's intentions. In any case, representing all semantic structures with the very same RL semantic graphs is, clearly, a large step toward amalgamation.

Natural language inferencing is not at all like that for mathematical logic. In propositional logic and predicate logic, a set of logical operators permit joining components into well-formed formulas, and inferencing proceeds from the (given) truth values of components to the (derived) truth values for whole statements. That is, in mathematical logic, inferencing proceeds from the bottom up. In natural language it is the other way around. Inferences are made from the top down, from the presumed truth value of the whole to the derived truth values of both explicit components and of inferred semantic structures. The fundamental reason is that the function of language is to communicate. Recall Sperber and Wilson's presumption of optimal relevance: the speaker

intends to make something manifest that is both optimally relevant and worth the addressee's while to process. It follows that the basic assumption we have as we process language is that what we hear or read is justified. Sperber and Wilson (1988:139) phrase this basic assumption in terms of 'faithfulness', asserting that "every utterance comes with a guarantee of faithfulness, not of truth. The speaker guarantees that her utterance is a faithful enough interpretation of the thought she wants to communicate."

Relational Logic uses the value t ('true') to indicate that the speaker's utterances are guaranteed to be justified (or faithful). For example, if someone states that "Max is tall and wears a mustache", we assign the entire statement the value t and deduce a) that 'Max is tall', and b) that 'he wears a mustache' also have the value t. The truth values in RL are t ('true' or 'yes'), f ('false' or 'no'), and t/f ('indeterminate' or 'unknown' or 'don't know'). The necessity for at least three values is simply illustrated by the problem posed in answering such questions as: "Do they grow a lot of coffee in Venezuela?" to which a truthful answer, based on all the knowledge at one's disposal, could be "yes", "no", or "I don't know." RL has general, relational mechanisms for inferring logical and structural relationships among sentences and sentence components, for inferring valid, meaningful sentences and semantic components from speakers' utterances, and for inferring valid conclusions from a posed query relative to a knowledge base of natural language messages. In the next several paragraphs we examine some of these inference mechanisms.

To illustrate how natural language deduction works, we consider first the implicative and negative implicative verbs (Karttunen, 1971). For implicative verbs (like *manage*), if the whole sentence is true, then so is the complement of the verb. The truth of *John managed to let out the cat* implies that *John let out the cat*. On the other hand, if it is false that *John managed to let out the cat*, then it is false that *John let out the cat*. Finally, if we do not know whether or not *John managed to let out the cat*, then we do not know whether *John let out the cat*. For negative implicative verbs (like *forget*), if the whole sentence is true, then the complement of the verb is false; if the whole sentence is false, then the complement is true; and, if the whole sentence is indeterminate, then the truth of the complement is unknown. For example, if it is true that *John forgot to phone his mother on her birthday*, then it is false that *John phoned his mother on her birthday*; if it is false that *John forgot to phone his mother on her birthday*, then it is true that *John phoned his mother on her birthday*; and, if we do not know whether *John forgot to phone his mother on her birthday*, then we do not know whether or not *John phoned his mother on her birthday*. Truth tables for implicative and non-implicative verbs and their complements are given in 13) and 14). The truth table for a negative element (like *not*) and what it modifies is given in 15). This table is just like that for the negative implicatives (hence, the 'negative' label).

13)

Implic.		Comp.
t	→	t
f	→	f
t/f	→	t/f

14)

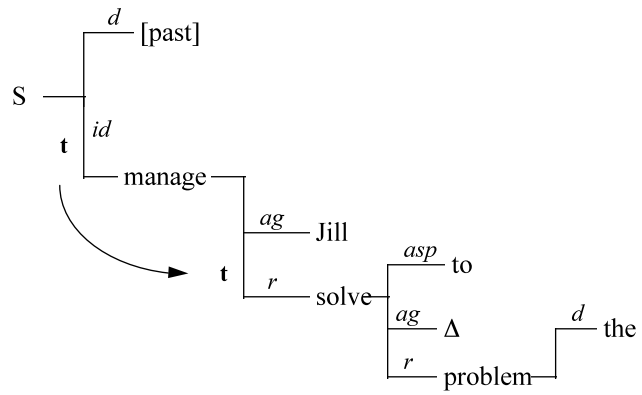
Neg.-Implic.		Comp.
t	→	f
f	→	t
t/f	→	t/f

15)

Neg.		Mod.
t	→	f
f	→	t
t/f	→	t/f

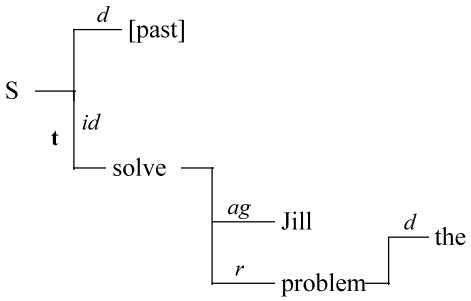
The semantic graph in 16) of *Jill managed to solve the problem* shows the aapplication of the truth table in 13) for implicative verbs. The truth value of the situation, indicated by **t** on the *id*-operator edge, is inferred to be true for the circumstance headed by *solve.*

16)



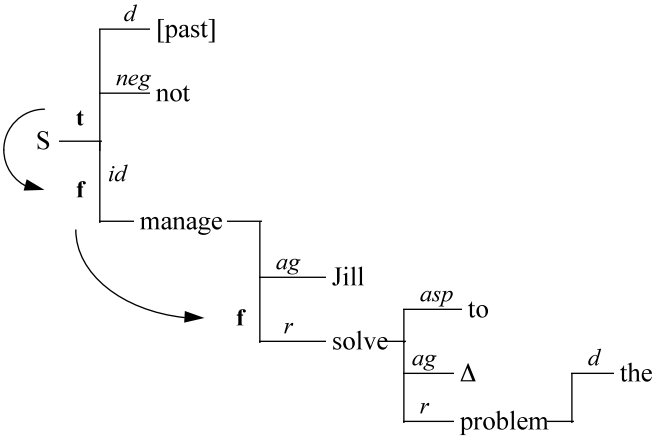
A circumstance inherits the situators of the situation in which it is embedded. This circumstance inherits, therefore, the past tense situator of *manage*, and we infer the truth of the sentence *Jill solved the problem*, whose semantic graph is given in 17).

17)



Truth values propagate through complex semantic structures. Sentences with a negative and an implicative and those with a negative and a negative implicative give straightforward inferences. For example, *Jill didn't manage to solve the problem*, whose semantic graph is shown in 18), implies that *Jill didn't solve the problem*. Similarly, *John didn't forget to phone his mother* implies that *John phoned his mother*.

18)



There are, of course, many verbs (like *believe*, *imagine*, and *hope*) that are non-implicatives. For non-implicative verbs, the truth value of their complements is indeterminate (t/f) regardless of the truth value of the dominating semantic component. For example, whatever the truth value of *Bob believed that Jill managed to solve the problem*, we do not know whether or not *Jill managed to solve the problem* or whether *Jill solved the problem*. In fact, once a t/f-value is encountered in any complex semantic structure, at the topmost situation or anywhere below it, then all truth values from that

point in the structure are indeterminate. That is, neither a t-value nor an f-value can ever be inferred when the dominant value is a t/f-value.

3.2.3 Pragmatics and Discourse

From a language understanding perspective, any theory of pragmatics and any theory of discourse are just subparts of a theory of communication, a way of accounting for how it is that a speaker's intentions are formulated, organized, interpreted, and understood. Grice proposed a theory of cooperation among the participants in a discourse or dialog (Grice, 1975, 1981.) Sperber and Wilson's theory of relevance encompasses and supersedes Grice's pioneering work and attempts an overall framework for understanding verbal (and other) communication. The task for pragmatics and discourse analysis is to take the explicit linguistic semantics representation of utterances and to derive the intended interpretation. Given our goal of amalgamation, we can think of this as a conceptual or propositional semantics representation. While pragmatic analysis makes use of real-world information at every turn, there is, obviously, no guarantee that the interpretation arrived at actually obtains in the real world.

(1) Pragmatics

Pragmatics is concerned with the use and interpretation of language in context. Determining the intended interpretation of language requires real-world knowledge; and, typically, it requires a lot of knowledge. The sorts of information that language users bring to bear include information about the basic vocabulary, say, the most commonly used ten-to-twenty thousand English words in their most common senses. This knowledge is shared with the larger English-speaking community. Alongside this general stock of information, there is a seemingly endless array of domain-specific vocabularies and of special senses of common words. Jargons include those associated with occupations (e.g., law, stock trading, military, medicine, farming, academe), technologies (automobiles and trucks, computing, airplanes, and economics), individual companies and regions, and sports and outdoor activities (baseball, tennis, fly fishing, duck hunting, and mountain climbing). One should not ignore, either, the role of history, since terminology varies and changes over time, and what is well known and understood in one era may be virtually unknown to most people in another. (Even widespread vocabulary of recent vintage can quickly fall into disuse. For instance, key-punch cards, key-punch machines, and key-punch operators are no longer evident, and knowledge of this phase of the history of computing is rapidly fading.) Knowledge of vocabulary presupposes that one knows how to use it. That is, it presupposes that one knows how the vocabulary is used to organize and characterize events into typical scenarios and topics. It is, in fact, one's experience and expectations about "goings on in the world", including their description, that enable language interpretation to proceed at all.

How is all this knowledge organized? Certainly, taxonomies and concept hierarchies play a role. Knowing that X "is covered by" Y is one basic kind of vocabulary knowledge. So is knowing that X is associated with Y, X is a part of Y, and X is the opposite or antonym of Y. Concept hierarchies are "tangled", since a given term can participate in a number of relationships. The organization of vocabulary terms in *Roget's Thesaurus*, with its relatively flat and shallow hierarchy of many categories, is very much in accord with prototype theory (See, for example, Rosch, 1978; Tversky, 1986). Not very surprisingly, basic-level (English) vocabulary items like *dog*, *bird*, *chair*, *tree*, *water*, *dirt*, *white*, *red*, *talk*, *walk*, *eat*, *drink*, and *sleep*, which are eminently suited for the human scale of experience and interaction with the environment, are in the middle of the hierarchy. Basic

terms like *chair* typically have superordinate terms (such as *furniture*) and subordinate terms (like *rocking chair* and *recliner*). Basic-level terms typically have a best exemplar, a "cognitive reference point" that forms the basis for inferences (Rosch, 1983). For example, the robin is about the size of the prototype bird. Hence, a chickadee is a small bird, and a raven is a large bird. Then, too, terms can be modified in such a way as to cancel temporarily some of their prototypical attributes. Thus, there is no contradiction in speaking of a dead bird, in noting that some birds are flightless, or in using the word *bird* to designate the figurine of a bird. Basic-level terms do not necessarily have a single prototype. The word *tree*, not in any case a botanical concept, has three basic prototypes--broadleaf, conifer, and palm. Depending on one's life experience, one or two of the three types might be more cognitively salient than the other two. One recent and important effort to organize and categorize English vocabulary is WordNet, an on-line lexical reference system organized in terms of lexical concepts that reflect human cognition and lexical memory. The overall organization of the WordNet database is described in G. Miller, et al (1990); nouns and lexical inheritance properties in G. Miller (1990); adjectives in Gross and K. Miller (1990); verbs in Fellbaum (1990); and computer implementation in Beckwith and G. Miller (1990).

Whatever one knows about words and what goes on in the world, it is nonetheless true that new words and old words in new uses are constantly encountered. What language users do to determine the meaning of new items must be explained by linguistics and functionally duplicated by a NLP system, which will also be confronted with "unknown words". At least in part, we can capitalize on the fact that speakers and writers are sensitive to the expectations of addressees, usually provide implicit definitions of items that they believe their audience will not be familiar with, and fail to give descriptive clues about items they believe their intended audience should be familiar with. For example, given the expectation that virtually every adult American knows that Canada is the country immediately to the north of the United States, it is highly unlikely that a news story would contain "In Canada, the country to the north of the United States, ..." On the other hand, it is quite likely that a news story about Burkina Faso would identify it as a West African country-- on the reasonable assumption that most Americans do not know what Burkina Faso is, let alone where it is. Appositives are one of the favorite mechanisms for introducing the meaning of unknown words: "Abra Cadabra, the Foreign Minister of ...," "Abra Cadabra, the oil-rich province of ...," "Abra Cadabra, the new software program from ..." The point is that NLP systems, like ordinary language users, must exploit whatever is available in any discourse to ferret out the meaning and categorizations of new items.

Much of the recent work in cognitive grammar seeks to explain the pervasiveness of metaphor and metonymy in language. (See especially Lakoff and Johnson (1980), Lakoff (1987), Johnson (1987), Mac Cormac (1985). In essence, the view that has emerged is that human thought and reason have a rich conceptual structure, one that necessarily and characteristically supports and demands the use of figurative language. As Lakoff puts it (1987:xiv):

- Thought is *embodied*, that is, the structures used to put together our conceptual systems grow out of bodily experience and make sense in terms of it; moreover, the core of our conceptual systems is directly grounded in perception, body movement, and experience of a physical and social character.

- Thought is *imaginative*, in that those concepts which are not directly grounded in experience employ metaphor, metonymy, and mental imagery--all of which go beyond the literal mirroring, or *representation*, of external reality. It is this imaginative capacity that allows for "abstract" thought and takes the mind beyond what we can see and feel....
- Thought has *gestalt properties* and is thus not atomistic; concepts have an overall structure that goes beyond merely putting together conceptual "building blocks" by general rules.

Basic-level concepts like chairs, birds, and dogs have gestalt properties. While such concepts clearly have internal structure, the wholes seem to be altogether more cognitively salient and basic than the parts.

Martin, 1992, presents a proposal for implementing a computer-based capability for understanding metaphors. Martin's approach provides an explicit representation of conventional metaphors and a mechanism (metaphor maps) for recognizing them. For example, to understand "How do I kill this [computer] process?", it is necessary to map "kill" onto terminate. Similarly, "How do I open up this database application?" requires that "open up" be mapped onto "start."

Another vexing problem is that of polysemy (related word senses), which is to be distinguished from homonymy. Homonomy refers to cases of accidental identity, such as the word *pen* (writing instrument or enclosure). Similarly, *bank* has a number of distinct and homonymous meanings, as in such noun compounds as *river bank* and *savings bank* and such verb uses as *bank money*, *bank a plane*, *bank a fire*. The various meanings of *bank* are not nowadays felt to be systematically related, although they may have been at an earlier time. For polysemy, where the various senses of a word do seem to be systematically related, the major research issue is the establishment of criteria for determining the number of senses. Two criteria that indicate different senses are 1) the systematic use of a word in different relational operator configurations and 2) the systematic co-occurrence of a word with words from different parts of the concept hierarchy. From a Relational Logic perspective, in the first case the arc labels are different, while in the second, the nodes characteristically have different "fills". For example, two senses of *open* are indicated for *John opened the door* and *The door opened*, since in the first use the relational operators are cause and undergoer , with relational structure *open*(*c*:X, *u*:Y), while, for the second, the single operator is just the undergoer, and we have *open*(*u*:X). Similarly, for *teach* we note such examples as *Mary taught algebra*, with the relational structure *teach*(*c*:X, *r*:Y), *Mary taught Bill*, with *teach*(*c*:X, *u*:Y), and *Mary taught Bill algebra*, with *teach*(*c*:X, *u*:Y, *r*:Z). Our pragmatic expectations about what can fill a given node largely determine ambiguity resolution. In the preceding example, the knowledge that *algebra* is a subject and that *Bill* is a personal name is the basis for determining which of the two transitive relational structures is intended. For prepositions, which typically have many senses, it is often the object of the preposition that enables ambiguity resolution. For instance, in *on Tuesday*, and *on the table*, the temporal and (one of the) locative senses of *on* are disambiguated by the choice of object. Brugman (1981) examined the large range of senses for *over*, including its use as a preposition, particle, and adverb. (This work is discussed at length in Lakoff, 1987:418ff.) The senses of *over* include 'above-across' (*fly over*), 'above' (*hang over*), 'covering' (*cloud over*), 'reflexive' (*turn over*), 'excess' (*spill over*), and various metaphorical senses (*oversee*, *overlook*). Brugman shows that the numerous senses of *over* are, indeed, systematically related.

Disambiguating the uses of *over* in these few examples depends on the category of what *over* modifies: *fly*, *hang*, *cloud*, *turn*, and *spill* are in different parts of the concept hierarchy.

The senses and uses of words are not static. New uses occur with great frequency and are to be expected. For example, recent new uses of the word *word* include *word processor* (a computer application program for authoring and editing text), *Microsoft Word* (a word processing product), *Word Grammar*, a theory of grammar, and WordNet, an on-line lexical database. Evidently, new word senses are easily acquired in context, and disambiguation depends on noting part of speech, relational semantic structure, and the co-occurrence of words from different parts of the concept hierarchy (including specific items, as in these examples). For both people and for NLP systems, then, learning new vocabulary and new senses of existing vocabulary items is a major concern. Yet, linguistics has paid almost no attention to this area.

(2) Discourse

A theory of discourse understanding must encompass ***interactive*** dialogs, short text messages (including memos and letters), narratives, and extended texts of the sort that typify expository writing. The theory of discourse structure advanced by Grosz and Sidner (1986) has been particularly influential. Grosz and Sidner propose that discourse structure is composed of three distinct, but interrelated, components: linguistic structure, intentional structure, and attentional state. Viewed as linguistic structure, a discourse consists of an assemblage of discourse segments. The segments consist of utterances (written or spoken), which are the basic linguistic elements. While the organization of discourse segments is largely linear and hierarchical (since discourses consist for the most part of topics and subtopics), the discourse model also provides for segments embedded in other segments, and for asides, interruptions, flashbacks, digressions, footnotes, and the like. The intentional structure component accounts for the purposes and aims of a discourse. Each discourse segment must have a purpose. Further, the originator of a segment must intend that the recipient(s) recognize the intention. As Grosz and Sidner say: "It is important to distinguish intentions that are intended to be recognized from other kinds of intentions that are associated with discourse. Intentions that are intended to be recognized achieve their intended effect only is the intention is recognized. For example, a compliment achieves its intended effect only if the intention to compliment is recognized ... (1986:178)" Although apparently formulated quite independently, Grosz and Sidner's insistence that discourse intentions be manifest agrees wholly with Sperber and Wilson's theory of Relevance. Much recent work (e.g., Asher and Lascarides, 1994) continues to investigate discourse intentions. The attentional state, the third component of Grosz and Sidner's discourse model, refers to the 'focus spaces' that are available to the participants in a discourse as the discourse unfolds. "[The attentional state] is inherently dynamic, recording objects, properties, and relations that are salient at each point in the discourse. ... changes in attentional state are modeled by a set of transition rules that specify the conditions for adding and deleting spaces (1986:179)." The development of a framework for modeling the attentional state, called centering, has been developed since the mid-1980s by Grosz, Joshi, and Weinstein. (See Grosz, Joshi, and Weinstein, 1995). Another important issue is how, in third person narratives, the reader (or listener) recognizes that the point of view is shifting from the narrator to one or another of the characters. Wiebe, 1994, discusses at length the mechanisms that underlie understanding and tracking a narrative's psychological point of view.

While Grosz and Sidner discuss only linguistic utterances and linguistic structure as a discourse component, it is obvious that discourses can contain many other sorts of elements and segments (which may be linguistic in part). Among them are, in spoken discourse, virtually anything that can be pointed to as a deictic element. In written texts these elements and segments include figures, drawings, pictures, graphs, and tables. In the newer discourse structures that use electronic multimedia, various graphics, video, and sound elements can be included in a discourse. That is not to say that everything one might link together on-line and electronically, no matter how sensibly, forms a discourse. Quite the contrary. For instance, in preparing an on-line version of a technical manual, one could provide a link from every mention of every technical term to its definition in a technical glossary. None of these links or definitions would be part of the discourse; they would be included as a convenience, merely for ready reference.

Within a discourse segment, the discourse coherence relations among the situations are often implicit and involve such notions as cause, consequence, claim, reason, argument, elaboration, enumeration, before, and after. (See Sanders, Spooren, and Noordman, 1993, for a recent discussion and proposal.) On the other hand, many transitions within a discourse structure, especially changes and transitions from one segment to another, are often made overt through the use of "clue word" or "cue phrase" expressions that provide information at the discourse level. These expressions include *incidentally*, *for example, anyway, by the way*, *furthermore, first, second, then, now, thus, moreover, therefore, hence, lastly, finally, in summary,* and *on the other hand.* (See, for example, Reichmann, 1986, for extended discussion of these expressions, and Hirschberg and Litman, 1993, on cue phrase disambiguation.) Anaphora resolution has also been the subject of much work in NLP. Representative treatments are given in Hobbs, 1978, Ingria and Stallard, 1989, Lappin and Leass, 1994, and Huls, Bos, and Claasen, 1995. In NLP, discourse referents (i.e., discourse anaphora) have themselves been much studied (see Webber, 1988, and Passonneau, 1989). Lastly, ellipsis is yet another topic that is at once a prominent feature of discourse structure and very important to language understanding. A recent treatment is found in Kehler, 1994.

Needless to say, no language understanding system currently available (or "on the drawing board") has anything even remotely close to a complete implementation of the linguistic elements outlined above. The full range of linguistic and cognitive phenomena to be covered is so incredibly complex that it is arguable whether linguistic theory and its near relatives that treat communication are at all mature enough to support the development of a semantic representation and inferencing capability satisfactory for linguistic semantics and for pragmatic interpretation in context. Nevertheless, the development of NLP capability is proceeding at a rapid pace, sometimes in reasonable accord with one or another linguistic theory, but often exploiting representation schemes, analysis methods, and inferencing techniques developed in computer science for other purposes.

4. Linguistically-based and statistically-based NLP

The purpose of an NLP system largely determines the approach that should be taken. Broadly speaking, there are two major approaches one can take to NLP implementation, namely, linguistically (i.e., knowledge) based and statistically based. For text analysis, if one's purpose is to build a system for very accurate, detailed information extraction, an objective that requires language understanding, then only a linguistically-based approach will serve. A comprehensive linguistically-based approach requires, however, full lexical, morphological, syntactic, semantic, pragmatic and discourse components. These are not

easy to come by. Less ambitious goals for text analysis--for instance, finding out what very large numbers of documents are "about" – can make excellent use of statistical methods.

Singular value decomposition (SVD), one such statistical method (See Berry, Dumais, and O'Brien, 1995), is a promising technique for achieving conceptual ***indexing***. Conceptual indexing, which correlates the combined occurrence of vocabulary items with individual text ***objects*** (typically a few paragraphs in length), enables querying and retrieval of texts by topic – in accord with what they are "about". The objective is not language understanding; rather, it is to achieve robust text indexing and information retrieval that does not depend on the presence in a text of particular vocabulary items.

SVD is, like many other statistical techniques, a practical means for investigating and analyzing large corpora. The goals for large corpora analysis are many. In addition to conceptual indexing, they include: finding instances (in context) of interesting and/or rare language phenomena; determining the frequency with which language phenomena occur; discovering linguistic rules, constraints, and lexical items; and constructing bilingual dictionaries and/or ascertaining translation quality (by aligning texts, one a translation of the other). Two special issues of *Computational Linguistics* (March 1993, June 1993) were devoted to large corpora analysis. (See especially the introduction by Church and Mercer, 1993.) In sum, from a linguistic point of view, statistical techniques are not ends in themselves, but are tools to get at knowledge about language or the world. Large corpora analysis techniques, in particular, give several different sorts of results of direct interest to linguists. Among them are: 1) using conceptual indexing of a large number of short texts (or long texts segmented into suitable "chunks"), to select texts on particular topics for some linguistic purpose or other, 2) culling example sets of some linguistic phenomena from very large collections of text, and 3) finding bilingual equivalents of lexical items in (presumably) equivalent contexts.

For natural language understanding applications per se, statistical methods can augment--and complement--rule-based systems. For instance, since any system will, in operational use, repeatedly encounter new (i.e., 'unknown') lexical items, an automatic part-of-speech tagger can be used to make a best guess as to the correct part of speech of the unknown item. Further, no semantic lexicon will be complete, either; and an automatic semantic tagger can make a best guess as to the category in which an unknown term is being used (minimally, "person", "place", or "thing") or can suggest the sense in which a known word is being used. For instance, if we encounter "He introduced that idea several lectures ago", a semantic tagger could suggest that "lectures" is being used as a temporal expression. (The literature on statistical methods for language analysis is burgeoning. See, for example, Kupiec, 1992; Charniak, 1993; Pustejovsky , Bergler, and Anick, 1993: Merialdo, 1994; Brill, 1995; Roche and Schabes, 1995, Mikheev, 1996.)

5. Controlled Language Checking

To meet the needs of users of technical documentation, especially those whose native language is not that in which the materials are written, a highly desirable goal is to restrict the vocabulary and grammatical structures to a subset of that which would ordinarily occur. Codifying the restrictions systematically defines a controlled language standard. One of the best-known is Simplified English, developed by AECMA (Association Europeene des Constructeurs de Materiel Aerospatial) [AECMA, 1986, 1989], and mandated by the Air Transport Association as the world-wide standard for commercial aircraft maintenance manuals. The general English vocabulary allowed in Simplified English (SE) is about

1500
words, only about 200 of which are verbs. Except for a few common prepositions, each of these words is to be used in one, and only one, prescribed meaning. Aerospace manufacturers are to augment this highly restricted core vocabulary with technical terms. (Boeing adds over 5,000.)

SE grammatical and stylistic restrictions are wide ranging. For example: the progressive verb forms and the perfective aspect are not allowed; the past participle is allowed only as an adjective; the passive voice is not allowed in procedures (and is to be avoided in descriptions); singular count nouns must be preceded by a determiner; noun groups should not contain more than three nouns in a row (long technical terms should use a hyphen to join related words); sentences in procedures should not be longer than 20 words, those in descriptions no longer than 25 words; verbs and nouns should not be omitted to make sentences shorter; instruction sentences cannot be compounded, unless the actions are to be done simultaneously; paragraphs should have no more than six sentences, and the first sentence must be the topic sentence; warnings should be set off from other text.

While manuals that conform to SE are easy to read, the many restrictions make writing them extremely difficult. To meet the need of its engineer writers to produce maintenance manuals in SE, Boeing developed a syntax-based Simplified English Checker (SEC), (Hoard, Wojcik, and Holzhauser, 1992). Since its introduction in 1990, the Boeing SEC, which contains a grammar of roughly 350 rules and a vocabulary of over 30,000 words, has parsed about 4 million sentences. Random sampling shows that the SEC parses correctly over 90% of the sentences it encounters, detects about 90% of all syntactic SE violations, and reports critiques that are about 80% accurate (Wojcik, Harrison, and Bremer, 1993). The critiques suggest alternative word choices for non-SE terms (e.g., "verb error: result; use: cause") and note grammatical violations of the SE standard (e.g., too many nouns in a row). The SEC does not attempt to rewrite text automatically. Nor should it, since the author's intentions are completely unknowable to the SEC (and to any other syntax-only analysis system).

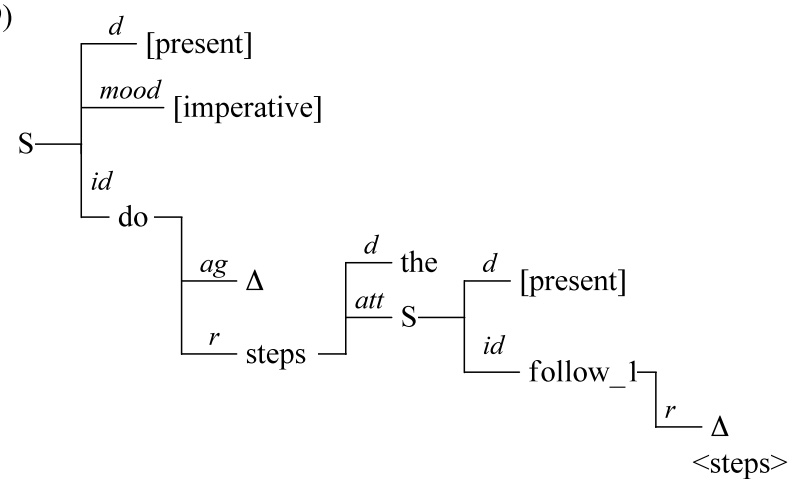
Even though Boeing's SEC is the most robust grammar and style checker of its type, it does not detect any semantic violations of the SE standard. To do so requires a meaning-based language checker, as depicted conceptually in Figure 2, that adds to the current SEC's Syntactic Analyzer and Syntactic Error Detector, an Initial Semantic Interpreter, a Word Sense Checker, and a Semantic Error Detector.

Currently in prototype, the Boeing meaning-based SEC adds knowledge of several types to the syntax-based checker. These include word-senses for all the words known to the system (most words have several senses), semantic hierarchies and categorizations (especially important for technical terms), a word sense thesaurus which indicates for every word sense whether it is sanctioned by the SE standard (and, if not, what alternative word is available), and semantic selection restrictions (including noun compound information and preferences that are specific to the application domain).

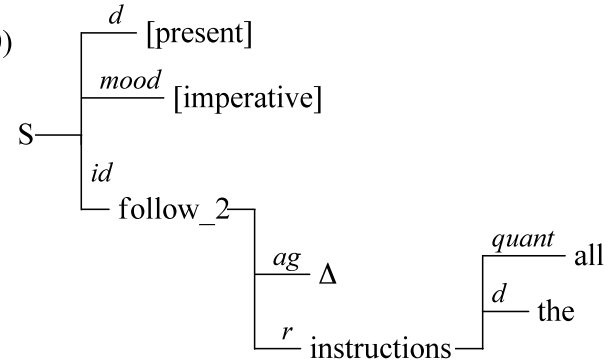
While the current SEC permits all the senses of an allowed SE word to pass unremarked, the meaning-based error detector makes full use of semantic graphs to find word-sense violations. For example, the verb *follow* is allowed in SE in the sense 'come after', but not in the sense 'obey'. The sentence *Do the steps that follow,* whose semantic graph is shown in 19), uses *follow* in the 'come after' sense (indicated in the graph by 'follow_1'). This is determined during the analysis of the sentence by noting that the

(implied) range complement of *follow* is *steps*. On the other hand, the sentence *Follow all the instructions*, uses *follow* in its 'obey' sense (indicated with 'follow_2' in the graph). This is determined during sentence analysis by noting that *follow* has an (understood) agent complement. (Note that the transitivity of *follow* does not determine the sense. The example sentence *A reception follows the concert* has the 'come after' sense of *follow*. Here, the semantic structure has *follow* with range and source complements, not with agent and range complements, as in 20). It is the difference in complement structures and in our expectations as to what can fill them that causes us to interpret *follow* as having one sense or the other.

19)



20)



Similarly, though the preposition *against* is restricted in SE to the sense 'in contact with', the current SEC does not detect the word-sense error in *Obey the precautions against fire*. But the meaning-based checker does, suggesting that *against* be replaced with the verb *prevent*. (A possible rewrite of the sentence is *Obey the precautions to prevent fire*.) Often, the meaning-based SEC can improve on the critique offered by the syntax-based SEC. For instance, the word *under* is not allowed in SE, and the current SEC suggests 'below, in, less than' as alternatives – advice that is only moderately helpful The meaning-based SEC is able to much better by determining the (apparent) intended sense of *under* and suggesting the one most appropriate alternative for consideration as the replacement.

The ability to do true meaning-based checking has far-reaching consequences for all

the application types listed at the beginning of this chapter. Obviously, being able to determine the sense in which a word is being used, as in the meaning-based SEC, will enable writers to produce materials that are far closer to that which is sanctioned by a restricted, controlled language standard like SE than ever before. But that is not the only gain. For instance, it is widely maintained that the quality of the inputs is the single most important variable in determining the quality of machine translation outputs. And, indeed, it appears that, for the foreseeable future, fully automatic machine translation will be possible only when the inputs are fully constrained by adhering to a restricted, controlled language standard. Then, too, with semantic interpretation of inputs and the ability of a system to negotiate intended meaning with users, natural language interfaces can be integrated with intelligent agent technology to provide general query capabilities (going well beyond the present ability to query a single database). Further, the ability to provide declarative knowledge bases in controlled English, for both modeling and process descriptions, and to translate the descriptions into machine-sensible underlying formalisms will greatly expand the speed with which such systems can be implemented. All of this is just to say that analyzing and controlling the meaning of the inputs to an NLP application provides a level of ambiguity resolution that identifies, reduces, and eliminates ambiguity to a degree that will give high confidence in the functioning of any system of which it is a component.

While we are clearly only at the beginning of the effort to fashion computer systems that understand language, it should be obvious that a linguistics whose objectives are broadened to include language understanding has a large role to play. Conversely, NLP has much to offer linguistics, for real-world applications test even the most comprehensive theories of language understanding to the limit. Boeing's prototype meaning-based Simplified English Checker is an early example of such an application, one whose worth will (or will not) be borne out in actual production use when the first few million sentences submitted to it are interpreted and critiqued.

Theoretical and Computational Linguistics: Toward a Mutual Understanding

Samuel Bayer John Aberdeen John Burger
Lynette Hirschman David Palmer Marc Vilain
The MITRE Corporation

1. Introduction

The nature of computational linguistics (CL) has changed radically and repeatedly through the last three decades. From the ATN-based implementations of transformational grammar in the 1960s, through the explicitly linguistics-free paradigm of Conceptual Dependencies,[1] to the influence and applications of 1980s-era unification-based frameworks, CL has alternated between defining itself in terms of and in opposition to mainstream theoretical linguistics. Since the late 1980s, it seems that a growing group of CL practitioners has once more turned away from formal theory. In response to the demands imposed by the analysis of large corpora of linguistic data, statistical techniques have been adopted in CL which emphasize shallow, robust accounts of linguistic phenomena at the expense of the detail and formal complexity of current theory. Nevertheless, we argue in this chapter that the two disciplines, as currently conceived, are mutually relevant. While it is indisputable that the granularity of current linguistic theory is lost in a shift toward shallow analysis, the basic insights of formal linguistic theory are invaluable in informing the investigations of computational linguists; and while ***corpus***-based techniques seem rather far removed from the concerns of current theory, modern statistical techniques in CL provide very valuable insights about language and language processing, insights which can inform the practice of mainstream linguistics.

There are two forces driving the evolution of this brand of CL, which we will call *corpus-based CL*, that we hope to emphasize in the sections to follow. The first is that the complexity and power required to analyze linguistic data is *discontinuous* in its distribution. Coarsely put, we have seen over and over that the simplest tools have the broadest coverage, and more and more complexity is required to expand the coverage less and less. Consider the place of natural language as a whole on the Chomsky hierarchy, for instance. Chomsky (1956) demonstrated that natural language is at least context-free in its complexity, and after a number of failed proofs, it is now commonly agreed that natural language is strongly and weakly trans-context-free (Shieber 1985, Kac 1987, Culy 1985, Bresnan *et al.* 1982). Yet what is striking about these results is both the relative infrequency of constructions which demonstrate this complexity and the increase in computational power required to account for them. For example, the constructions which are necessarily at least context-free (such as center embedding) seem fairly uncommon in comparison with constructions which could be fairly characterized as finite state; the constructions which are necessarily trans-context-free are even fewer. In other words, a large subset of language can be handled with relatively simple computational tools; a much smaller subset requires a radically more expensive approach; and an even smaller subset something more expensive still. This observation has profound effects on the analysis of large corpora: there is a premium on identifying those linguistic insights which are simplest, most general, least controversial, and most powerful, in order to exploit them to gain the broadest coverage for the least effort.

The second force driving the evolution of corpus-based CL is the desire to measure progress in the field. Around 1985 a four-step paradigm began to evolve which has motivated

a wide range of changes in the field of CL; among these changes is the reliance on initial broad, shallow analyses implied by the discontinuous nature of linguistic data. This methodology, which we describe and exemplify in detail below, is responsible for introducing quantifiable measures of research progress according to community-established metrics, and has led to an explosion of new ways to look at language.

These two forces constitute the first half of our story. First through history and then through examples, we will illustrate how substantial advances have been made, and measured, in a paradigm which favors broad coverage over fine-grained analysis. We will show that the corpus-based CL commitment to evaluation has led to the insight that simple tools, coupled with crucial, profound, basic linguistic generalizations, yield substantial progress. But the story does not end there. It is not simply that linguistics informs the corpus-based, evaluation-based paradigm; the reverse is also true. We believe that the demands that large corpora impose on linguistic analyses reveal many topics for inquiry that have not been well explored by traditional linguistic methods. Abney (1996) argues that modern theoretical linguistics poorly accounts for, or fails to account for, a range of issues including the graded nature of language acquisition, language change, and language variation, as well as disambiguation, degrees of grammaticality, judgments of naturalness, error tolerance, and learning vocabulary and grammar on the fly by mature speakers. Abney further contends that current corpus-based CL approaches respond to exactly those considerations which compromise current theoretical techniques. In many cases, these approaches are born of researchers' frustration with faithful implementations of their theories: gaps in coverage and expressiveness, intolerable ambiguity, an inability to model graded distinctions of grammaticality or likelihood of use. It is with an interest in these less-commonly-asked questions that we invite you to read the narrative to follow.

2. History: Corpus-Based Linguistics

Corpus-based analysis of language is hardly a new idea. Following de Saussure, the American structural linguists, from Leonard Bloomfield (Bloomfield 1933) through Zellig Harris (Harris 1951), pursued an empirical approach to linguistic description, applying it to a large variety of languages (including Amerindian languages, Chinese, Korean, Japanese, and Hebrew) and a range of linguistic phenomena, predominantly phonology and morphology but also syntax and even discourse structure. For the structuralists, linguistic analysis required (1) an inventory of the distinct structural elements, (2) rules for the observed combinations of these elements, and (3) a procedure for discovering the units and their combinatorics via empirical observation based on systematic analysis of a corpus of utterances. The methods that the structuralists developed – distributional analysis and the study of co-occurrence data, decomposition of analysis into multiple layers of phonology, morphology, syntax and discourse, and automatable discovery of linguistic descriptions or grammars – underlie much of the current research on corpus-based methods.[2]

Reliance on, or reference to, naturally-occurring data is also taking hold in modern theoretical linguistics as well, and becoming more prevalent. Recent advocates and adherents include Birner (1994), Macfarland (1997), and Michaelis (1996); in many cases, researchers have relied on corpora to refute previously-proposed generalizations about linguistic constructions. We applaud this trend, certainly; but it is only part of the puzzle. When we talk about corpus-based linguistics today, we don't simply mean the consultation of a corpus in the course of linguistic research; we mean the commitment to robust, automatic analysis of this corpus, in as much depth as possible.

Given how old the goal of automated grammar discovery is, it is curious that it has

taken approximately fifty years to make real progress in this area, measured by systems that work and methodologies that can generate reasonable coverage of linguistic phenomena. The reasons for this are partly sociological, influenced by the methods in vogue in any particular decade, but mostly technological; they include

- the ready accessibility of computational resources (fast machines, sufficient storage) to process large volumes of data
- the growing availability of corpora, especially corpora with linguistic annotations (part of speech, prosodic intonation, proper names, bilingual parallel corpora, etc.), and increased ease of access and exchange of resources via the Internet
- a commercial market for natural language products, based on the increased maturity of computational linguistics technology
- the development of new tools, including both efficient parsing techniques (e.g., finite state transducers) and statistical techniques

For example, the statistical technique of ***Hidden Markov Models (HMM)*** revolutionized speech recognition technology (Rabiner 1989), making it possible to build robust speaker-independent speech recognizers. This technique, originally adopted by the engineering community for signal processing applications, has now been widely applied to other linguistic phenomena as well. A Hidden Markov Model provides a technique for automatically constructing a recognition system, based on probabilities provided by training data. An HMM consists of two layers: an observable layer and a hidden layer which is a Markov model, that is, a finite state machine with probabilities associated with the state transitions. For speech recognition, the observable layer might be the sequence of acoustic segments, while the hidden layer would be finite state models of word pronunciations, represented as phoneme sequences. Once the HMM is trained by presenting it with recorded speech segments together with transcriptions, it can be used as a recognizer for acoustic segments, to generate transcriptions from speech.[3]

Consider how this technique might be used in corpus-based CL. One of the most common tasks performed in corpus-based CL is *part-of-speech tagging*, in which lexical categories (that is, *part-of-speech tags)* are assigned to words in documents or speech transcriptions. It turns out that part-of-speech tags can be assigned in English with a very high degree of accuracy (above 95%) without reference to higher-level linguistic information such as syntactic structure, and the resulting tags can be used to help derive syntactic structure with a significantly reduced level of ambiguity. For part-of-speech tagging via HMMs, the observable layer is the sequence of words, while the hidden layer models the sequence of part-of-speech tags; the HMM is trained on documents annotated with part-of-speech tags, and the resulting trained HMM can be used to generate tags for unannotated documents.[4]

The success of Hidden Markov Models for speech recognition had a major effect on corpus-based language processing research. Here was a technique which produced an astounding improvement in the ability of a machine to transcribe speech; it was automatically trained by the statistical processing of huge amounts of data, both spoken transcribed data and written data, and its performance was evaluated on a never-before-seen set of blind test data. This success had profound commercial implications, making it possible, for example, to produce medium-vocabulary speech recognition systems that required no prior user training.[5] This technique was developed in the late 1960s and early 1970s; it made its way into the speech recognition community by the mid-1970s, and into commercial quality speech recognizers by the mid-1980s. In the process, these successes also created a new paradigm for research in computational linguistics.

3. History: Evaluation

Before we exemplify the corpus-based methodology and elaborate on its implications for theoretical linguistics, we want to define some terms and then trace the impact of evaluation on two communities: researchers in spoken language understanding and researchers in text understanding. When we speak of "understanding" in corpus-based CL, we intend a very limited, task-specific interpretation; roughly, we take a language processing system to have "understood" if it responds appropriately to utterances directed to it. For evaluated systems, the notion of an appropriate response must be defined very clearly in order to measure progress. Much of the work in this paradigm, then, consists of community-wide efforts to define the form and content of appropriate responses. Defining the system responses in this way has the added advantage that the internals of the language processing systems need not be examined in the process of evaluation; all that is required is the input data (speech or unanalyzed text) and the system's response. This sort of evaluation is known as a ***"black box" evaluation***, so called because the language processing system can be treated as an opaque "box" whose inputs and outputs are the only indications of its performance.

There have been two major efforts aimed at the evaluation of natural language systems: one in the area of speech recognition and spoken language systems, the other in the area of text-based language understanding. The speech recognition evaluations began as part of the ***DARPA*** speech recognition program, and from 1990-1995, the speech recognition evaluations were combined with evaluation of spoken language understanding (Price 1996). The focal point for the text-based evaluations has been the series of ***Message Understanding Conferences (MUCs)*** that have taken place every year or two since 1987 (Grishman and Sundheim 1995), with MUC-7 scheduled for the fall of 1997. This section briefly traces the evolution and history of corpus-based evaluation in these two research communities.

3.1. The Air Travel Information System (ATIS) evaluation

For speech recognition and spoken language, the push for evaluation came specifically from DARPA, the agency which funded much of the advanced research in this area. There was already a well-understood need for a corpus-based methodology, since speech recognizers rely heavily on the availability of (large amounts of) recorded speech with corresponding transcriptions and an evaluation function for automated training. Prior to the effort to evaluate the understanding aspects of spoken language interfaces, the measurement of speech recognition error had already been responsible for dramatic progress in the speech community.

To develop an automated approach to evaluating spoken language understanding, researchers chose the task of making air travel reservations. This task was chosen because it was familiar to many people, and thus promised a broad base of potential users who could act as experimental subjects for data collection to build such a language understanding system. Researchers limited the problem of language understanding for this task in some critical ways:

- Scope was limited to evaluation of spoken queries against a static database of airline information (as opposed to dynamic, unpredictable data for which the vocabulary might be unknown in advance)
- Interaction style was restricted to human-initiated queries only, since it was not clear how to automate evaluation of discourses where the language understanding system asked questions of the user as well
- The queries were restricted to a well-defined domain of reasonable size, in order to ensure that the database provided enough coverage to ensure useful interactions

- Evaluation was limited to a strict definition of answer correctness, based on comparison to the set of tuples to be returned from the database

In the context of this paradigm, users were presented with a task description (for instance, "You must make reservations to visit your grandmother in Baltimore, but you must stop in Boston overnight, and due to scheduling restrictions, you must leave on a Thursday"). The users' recorded utterances, along with their transcriptions and the correct database response, form the basis of the ATIS corpus.

Despite the limitations imposed by this strategy for evaluation of understanding, this work introduced some useful candidate standards: transcription as the basis for evaluation of speech recognition (borrowed from earlier work in the speech community); development of a methodology to evaluate answer correctness based on the tuples retrieved from a database; and an annotation method to distinguish *context-independent* queries from *context-dependent* queries for evaluating the understanding of multi-utterance sequences (the latter required that the system have some model of the preceding interaction to answer correctly). In addition, the researchers undertook a highly successful collaborative data collection effort, which produced a large body (20,000 utterances) of annotated data, including speech, transcriptions, database output, and timestamped log files of the interactions (Hirschman *et al.* 1992). Figure 1 shows an excerpt of a sample log file from the ATIS data collection efforts; it includes timestamped entries for receipt of speech input, transcription, translation into SQL (a standard database retrieval language), and system response.

These standardized evaluations resulted in a rapid decrease in the spoken language understanding error rate, showing that the research community was moving steadily towards solving the problem of getting machines to not only transcribe but also respond appropriately to what a user might say. Figure 2 plots error rate (log scale) against successive evaluation dates for the Spoken Language program, using figures for the best performing system in each

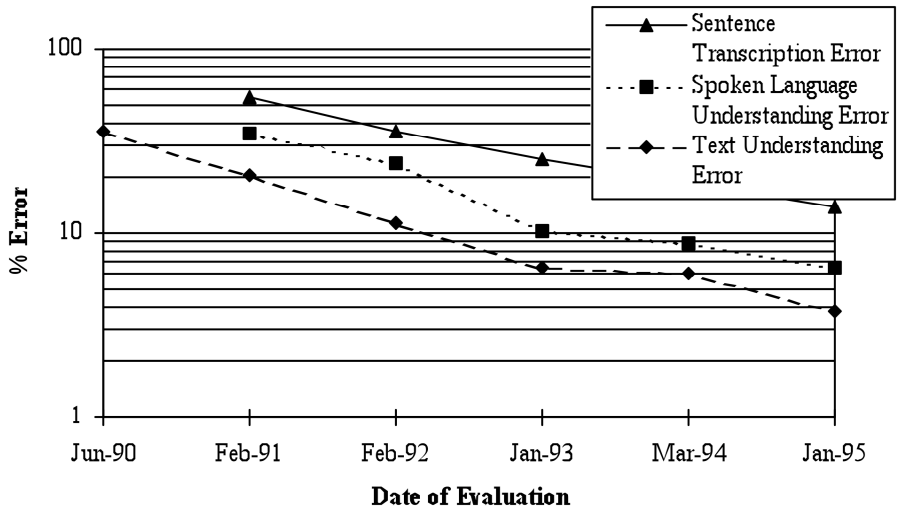


Figure 2: Rate of Progress in the DARPA Spoken Language Program for Context-Independent Sentences

evaluation, processing context-independent utterances.[6] The top line (highest error rate) is a measure

of sentences correctly transcribed (no understanding): a sentence is incorrect if it contains one or more word recognition errors. It is clearly much harder to get *all* the words right in transcribing a sentence, which is why the sentence transcription error rate tends to be high; the percentage of *words* incorrectly transcribed for the ATIS task had dropped to 2% by June 1995. The next highest line in Figure 2 represents spoken language understanding error rate, namely the percent of sentences that did *not* receive a correct response from the database, given the speech input. The lowest error rate is text understanding error rate – given a perfect transcription as input, this is the percentage of sentences that did not get a correct database response. Not surprisingly, the text understanding error is lower than the spoken language understanding error, since processing the speech input can result in a less-than-perfect transcription.

Most significant, and perhaps counterintuitive, is that the understanding error rate (that is, the percentage of incorrect database responses) is lower, at every point, for both speech and text, than the sentence transcription error rate (that is, the percentage of sentences with at least one word incorrectly transcribed). This means that it is easier for the system to understand a sentence (given our definition of "understanding") than to transcribe it perfectly. In other words, perfect transcription is by no means a prerequisite for language understanding, and systems which require perfect grammatical input are at a disadvantage in this task compared to systems which do not.

3.2. The Message Understanding Conferences (MUCs)

The message understanding conferences evolved out of the research community's need to share results and insights about text-based language understanding. The six MUC conferences to date have been responsible for:

- the creation of an automated evaluation methodology for understanding text messages, including annotated training and test data sets
- significant progress in building robust systems capable of extracting database entries from messages
- increasing the technology base in this area (participating groups have increased from six at the first conference in 1987 to sixteen at MUC-6 in November 1995)

The first three message understanding conferences (1987, 1989, 1991) avoided component-based evaluation, since there was little agreement on what components would be involved, and focused on a black box evaluation methodology. The evaluation was defined at a high level. Given documents as input, the system's performance was determined by the extent to which it could produce the appropriate set of templates describing who did what to whom in a particular topic area (e.g., terrorist attacks, or joint ventures). Figure 3 shows an excerpt of a sample document from the MUC-4 document set about terrorist activity, along with a simplified template.

```
CLANDESTINE, 20 NOV 89 (RADIO VENCEREMOS) -- [TEXT]...
SPANISH FOREIGN MINISTER LUIS YANEZ [AS HEARD] REPORTED
TODAY THAT SPAIN HAS SUSPENDED AID TO THE SALVADORAN
GOVERNMENT UNTIL THE KILLINGS OF THE SPANISH JESUITS, WHO
HAD BEEN LIVING IN OUR COUNTRY FOR YEARS, ARE RESOLVED...
INCIDENT: DATE                  - 20 NOV 89
INCIDENT: LOCATION              EL SALVADOR
INCIDENT: TYPE                  ATTACK
INCIDENT: STAGE OF EXECUTION    ACCOMPLISHED
HUM TGT: DESCRIPTION            "JESUITS"
HUM TGT: TYPE                   CIVILIAN: "JESUITS"
HUM TGT: NUMBER                 PLURAL: "JESUITS"
HUM TGT: FOREIGN NATION         SPAIN: "JESUITS"
HUM TGT: EFFECT OF INCIDENT     DEATH: "JESUITS"
```

Figure 3: Sample Document and Template Fragment from MUC-4

Over these three evaluations, the participants defined training and test corpora, a detailed set of guidelines specifying the content of the templates for the test domains, and an automated scoring procedure. The MUC evaluations also introduced some standard terminology into the evaluation paradigm. Scores were calculated by comparing system ***hypotheses*** against a human-generated ***key***, to produce numerical comparisons in terms of ***precision*** and ***recall.*** Precision is a measure of false positives; more precisely, it is the ratio of correct system answers to total system answers. Recall is a measure of false negatives; more precisely, it is the ratio of correct system answers to total answers in the key.

Once this basic methodology was established, the message understanding community was able to branch out in several directions, including evaluation of documents in multiple languages. The community has also been successful in decomposing the message understanding task into coherent subproblems: identifying proper names in text, identifying coreferring expressions, and gathering all the described attributes of an object or individual from its disparate references scattered through a text. Some of these tasks, such as identifying coreferring expressions, are still under development; nevertheless, it is clear that this functional decomposition, providing a layered application of linguistic knowledge, has been highly successful in defining the strengths and weaknesses of current technology, as well as opening new areas for research (e.g., tagging for coreference or for parts of speech in multiple languages).

4. Methodology

With this history in hand, we can now turn to a more detailed description of these corpus-based approaches to language processing. In the next few sections, we will outline and exemplify this family of approaches.

We can describe the corpus-based approach in four steps. First, we obtain and analyze linguistic data in the context of the task we choose (***information extraction***, named ***entity*** extraction, summarization, translation, etc.). Second, we hypothesize a procedure for producing the appropriate analyses. Third, we test this procedure, and finally, we iterate our procedure based on our evaluation. We will examine each of these steps in turn.

4.1. Step 1: analyze the data

Before we analyze data, we need to *have* data, in some computer-readable form. With the advent of powerful, inexpensive networked computing, this has become far more feasible than in the past. We now have corpora which encompass a wide range of data types (speech, text, multimedia documents), and increasingly, a range of languages. The community has also benefited considerably from well-organized efforts in data collection, exemplified by the multi-site ATIS and MUC efforts and the services and products provided by the ***Linguistic Data Consortium.***

Our analysis of the data must be guided by the task at hand, whether it be transcription, translation, search, summarization, database update or database query. The data provide us with the input; the structure of the task provides us with the output. For instance, in the ATIS task, the input is a speech signal and the output a database query. Our goal is to find a systematic, robust mapping from input to output. We may need to break that mapping down into many smaller steps, but we are still analyzing the input in terms of the desired output. In many cases, the form of the output is an annotated form of the input (for instance, a document augmented with the part of speech of each word); these annotations are drawn from a ***tag set*** of all possible tags for a given task.

4.2. Step 2: hypothesize the procedure

Based on our analysis of the data, we hypothesize a procedure that allows us to map from input to output. What matters is that this procedure can be implemented as a computer program. There are many approaches to choose among, from neural nets to stochastic models to rule-based systems. Some of these use explicit rules created by a human, some use machine learning, some are based on statistical processing. In general, the faster and more efficient the procedure, the more successful the procedure will be, because it will provide the opportunity for a greater number of iterations through the process (see Step 4 below).

4.3. Step 3: test the procedure

The corpus-based methodology uses data for two distinct purposes: to induce the analysis, and to provide a benchmark for testing the analysis. Our goal is to use this methodology to improve our performance on our chosen task. Therefore, the corpus-based method requires that there be a defined evaluation metric that produces a result, so that we can compare strategies or rule sets. If we cannot make this comparison, we have no reliable way of making progress, because we do not know which technique yields the better result.

We must choose our evaluation metric with care. We must believe that the evaluation metric has relevance to the task we are trying to perform, so that as our evaluation results improve, the performance of the system will improve. Furthermore, it is critical that we use new data (that is, data not used in the hypothesis phase) for evaluation, to ensure that we have created a system that is robust with respect to the kind of data we expect the system to have to handle. If we do not, we run the risk of designing strategies which are overly specific to the data we've used for training.

Finally, it is important to understand that the accuracy of our evaluation depends crucially on our accuracy in determining the "right" output, that is, our accuracy in creating the key. If any two humans performing the task in question can only agree on the "right" output 90% of the time, then it will be impossible for us to develop a system which is 95% accurate, because humans can't agree on what that means. So inter-annotator variability among humans sets an upper bound on the system accuracy we can measure.

4.4. Step 4: iterate

Once we evaluate our approach, we can use standard techniques to improve our results, such as a systematic machine learning approach, or iterative debugging, or regression testing. During the iteration, we can revisit any one of the previous steps. We may need to refine our tag set, our procedure – or even our evaluation. Depending on the scope of the problem, all of these may get revised in the course of research.

In the next two sections, we describe applications of this paradigm in the domain of text-based language understanding. We will attempt to emphasize these four steps, showing how progress can be made using this technique.

5. Example: Sentence Segmentation

While the corpus-based methodology has successfully pushed progress in many areas traditionally of interest to linguists, it has also revealed many new problems which are frequently overlooked or idealized away in theoretical linguistics, yet which are essential steps for large-scale processing of language. One example of such an area is the segmentation of linguistic data into sentences, a task which can be surprisingly complex.

Recognizing sentence boundaries in a document is an essential step for many CL tasks. Speech synthesizers produce output prosody based on a sentence model, and incorrect identification of boundaries can confuse them. Parsers, by definition, determine the structure of a sentence, and therefore depend on knowledge of sentence boundaries. However, dividing a document into sentences is a processing step which, though it may seem simple on the surface, presents a wide variety of problems, especially when considering different languages. For example, written languages with punctuation systems which are relatively impoverished compared to English present a very difficult challenge in recognizing sentence boundaries. Thai, for one, does not use a period (or any other punctuation mark) to mark sentence boundaries. A space is sometimes used at sentence breaks, but very often there is no separation between sentences. Detecting sentence breaks in written Thai thus has a lot in common with segmenting a stream of spoken speech into sentences and words, in that the input is a continuous stream of characters (or phonemes) with few cues to indicate segments at any level.

Even languages with relatively rich punctuation systems like English present surprising problems. Recognizing boundaries in such a written language involves determining the roles of all punctuation marks which can denote sentence boundaries: periods, question marks, exclamation points, and sometimes semicolons, colons, and commas. In large document collections, each of these punctuation marks can serve several different purposes in addition to marking sentence boundaries. A period, for example, can denote a decimal point, an abbreviation, the end of a sentence, or even an abbreviation at the end of a sentence. Exclamation points and question marks can occur within quotation marks or parentheses (really!) as well as at the end of a sentence.[7] Disambiguating the various uses of punctuation is therefore necessary to recognize the sentence boundaries and allow further processing.

In the case of English, sentence boundary detection is an excellent example of both the discontinuities discussed previously and of the application of the corpus-based methodology to solving a practical problem. Simple techniques can achieve a rather high rate of success, but incrementally improving this initial rate and recognizing the difficult cases can require a significant amount of linguistic knowledge and intuition in addition to a thorough analysis of a large corpus of sentences.

The first step in the corpus-based methodology, obtaining and analyzing the data, is

quite straightforward for this task; millions of sentences are readily available in many different languages. And while compiling and analyzing the data for some CL tasks involves linguistically-sophisticated knowledge about transcribing or translation, the key for the sentence boundary detection task can be constructed with virtually no linguistic training.

The second step in the methodology, hypothesizing the procedure to solve the problem, may seem simple at first. When analyzing well-formed English documents such as works of literature, it is tempting to believe that sentence boundary detection is simply a matter of finding a period followed by one or more spaces followed by a word beginning with a capital letter; in addition, other sentences may begin or end with quotation marks. We could therefore propose the following simple rule as our entire sentence segmentation algorithm:

```
sentence boundary =
     period + space + capital letter
     OR period + quote + space + capital letter
     OR period + space + quote + capital letter
```

It is only through actually testing this rule on real data (Step 3 of the methodology), that we become aware of the range of possibilities. In some corpora (e.g. literary texts) the single pattern above indeed accounts for almost all sentence boundaries. In *The Call of the Wild* by Jack London, for example, which has 1640 periods as sentence boundaries, this single rule will correctly identify 1608 boundaries (recall of 98.1%) while introducing just 5 false negatives (precision of 99.7%). It is precisely these types of results that led many to dismiss sentence boundary disambiguation as a simple problem. However, the results are different in journalistic text such as the *Wall Street Journal*. In a small corpus of the WSJ which has 16466 periods as sentence boundaries, the simple rule above would detect only 14562 (recall of 88.4%) while producing 2900 false positives (precision of 83.4%).

We can use this knowledge to improve our hypothesis iteratively (Step 4 of the methodology) and attempt to produce a better solution which addresses the issues raised by the real data. Upon inspection of journalistic text, we see that our simple rule fails in cases such as "Mr. Rogers," "St. Peter," and "Prof. Thomopoulos." We therefore modify our rule to include the case of an abbreviation followed by a capitalized word:

```
sentence boundary =
     period + space + capital letter
     OR period + quote + space + capital letter
     OR period + space + quote + capital letter
   UNLESS abbreviation + period + space + capital
```

This new rule improves the performance on *The Call of the Wild* by eliminating false positives (previously introduced by the phrase "St. Bernard" within a sentence), and both recall and precision improve (to 98.4% and 100%, respectively). On the WSJ corpus, this new rule also eliminates all but 283 of the false positives introduced by the first rule. However, this rule introduces 713 false negatives because many abbreviations can also occur at the end of a sentence. Nevertheless, precision improves to 95.1% because this augmentation produces a net reduction in false positives.

This last enhancement shows that recognizing an abbreviation is therefore not sufficient to disambiguate a period, because we also must determine if the abbreviation occurs at the end of a sentence. However, this problem ultimately illustrates the discontinuous nature of data in this area. An abbreviation like "St." is lexically ambiguous: it can mean "Saint,"

"street" or "state." Each of these interpretations has a different potential for ending a sentence, and disambiguation of these different interpretations is crucial for determining sentence boundaries. For instance, the current rule would correctly handle the use of "St." for "Saint" in the following example (from WSJ 11/14/91):

```
The contemporary viewer may simply ogle the vast wooded
vistas rising up from the Saguenay River and Lac St. Jean,
standing in for the St. Lawrence River.
```

However, it would not correctly handle this use of "St." for "street" (from WSJ 01/02/87):

```
The firm said it plans to sublease its current headquarters
at 55 Water St.  A spokesman declined to elaborate.
```

The simple techniques we've examined so far are not sophisticated enough to distinguish reliably among cases like these. Furthermore, these simple techniques rely on orthographic distinctions which are not always present. For text where case distinctions have been eliminated (as in e-mail, which is sometimes all lower case, or television closed captions, which is all upper case), the sentence task is noticeably more challenging. In the following example (also from the WSJ, 7/28/89), the status of the periods before "AND" and "IN" is not immediately clear, while in case-sensitive text their status would be unambiguous:

```
ALASKA DROPPED ITS INVESTIGATION INTO POSSIBLE CRIMINAL
WRONGDOING BY EXXON CORP. AND ALYESKA PIPELINE SERVICE CO.
IN CONNECTION WITH THE VALDEZ OIL SPILL.
```

These cases, like the "St." case, require an analysis of the linguistic text which is more sophisticated than the simple orthographic rules we've seen so far. Useful information about the document may include part-of-speech information (Palmer and Hearst 1997) morphological analysis (Müller 1980), and abbreviation classes (Riley 1989).

6. Example: Parsing

A second example of a practical application of this methodology can be seen in the recent history of parsing. Progress in corpus-based parsing began with the release of the Penn Treebank (Marcus *et al.* 1993), developed at the University of Pennsylvania between 1989 and 1992. The Treebank consists of 4.5 million words of American English, tagged for part-of-speech information; in addition, roughly half of the Treebank is tagged with skeletal syntactic structure (hence the name "Treebank").

The annotation of syntactic structure consists of a bracketing of each sentence into constituents, as well as a non-terminal labeling of each constituent. The guidelines for bracketing, as well as the choice of non-terminal syntactic tags, were designed to be theory-neutral. Consequently, the degree of detail in the bracketing is relatively coarse, as compared to the analysis one might see in a complete parse. Again, this annotation design was strongly influenced by a desire for high accuracy and high inter-annotator reliability. The syntactic tag set consists of fourteen phrasal categories (including one for constituents of unknown or uncertain category), as well as four types of null ***elements***.[8]

Here is an example sentence from Collins (1996), annotated for syntactic structure as in the Treebank:

```
 [S  [NP [NP John Smith]
         ,
         [NP [NP the president]
             [PP of IBM]]
         ,]
     [VP announced
         [NP his resignation]
         [NP yesterday]]
```

The existence of the Treebank has been essential in enabling the direct comparison of many CL algorithms, and much recent progress in a number of areas of CL can be credited directly to the Treebank and similar resources. This has been particularly true in parsing, a task for which it has been notoriously difficult to compare systems directly.

Progress in parsing has also been greatly aided by the development of several evaluation metrics. These measures were developed in a community-sponsored effort known as PARSEVAL (Black *et al.* 1991), with the goal of enabling the comparison of different approaches to syntactic analysis. All of these measures assume the existence of a reference corpus annotated for constituent structure with labeled brackets, as in the Treebank example above. This annotation is assumed to be correct, and is used as the key.

When a parser's hypothesis is compared to the key, several kinds of mismatches may occur:

- A bracketed constituent present in the key may not be present in the hypothesis
- A constituent may occur in the hypothesis but not correspond to anything in the key
- Two constituents from the key and the hypothesis may match in extent (that is, comprise the same words), but be labeled differently

The measures defined by PARSEVAL attempt to separate these various kinds of errors, and include the following:

- *Labeled recall* is the percentage of constituents in the key that are realized in the parser's hypothesis, in both extent and non-terminal label.
- *Labeled precision* is the percentage of constituents in the hypothesis that are present in the key, in both extent and non-terminal label.

Labeled recall accounts for the first type of error above, while labeled precision accounts for the second. Both of these measures require the label as well as the extent to be correct; that is, the third error type above is both a recall and a precision error. There are also versions of these measures, referred to as *un*labeled precision and recall, in which the non-terminal labels need not match. This weaker definition of correctness allows the evaluation of a system with a different set of non-terminals than the key. There is also another PARSEVAL measure that disregards labels:

- *Crossing brackets* is the number of constituents in the hypothesis that have incompatible extent with some constituent in the key, i.e., which overlap with some key constituent, but not in a simple substring/superstring relationship. A typical crossing bracket violation arises if the key contains the bracketing

  `[large [animal preserve]]`,

  but the hypothesis brackets the string as

  `[[large animal] preserve]`.

Crossing brackets may be expressed as a percentage of the constituents in the hypothesis, similarly to precision and recall, but is more often a simple count averaged over all sentences in the test corpus. In particular, *zero crossing brackets* is the percentage of sentences with no such extent incompatibilities.

We can use our Treebank sentence from above to provide examples of each of these measures. The bracketed sentence is reproduced below, followed by a candidate parse hypothesis.[9]

```
Key:
[S  [NP [NP John Smith]
        ,
        [NP [NP the president]
            [PP of IBM]]
        ,]
    [VP announced
        [NP his resignation]
        yesterday]
```

The hypothesis has one crossing bracket error, due to the boundary violation between the hypothesis constituent `[NP president of IBM]` and the key's `[NP the president]`. The key has eight constituents, the hypothesis nine. Six of the hypothesis' constituents match constituents in the key exactly, and thus labeled precision is 75% (6/8), while labeled recall is 67% (6/9).

A recent breakthrough in parsing that relied critically on resources such as the Penn Treebank and the evaluation mechanisms introduced by PARSEVAL was the work of David Magerman (Magerman 1994). Magerman used probabilistic decision trees, automatically acquired from the Treebank and other annotated corpora, to model phenomena found in the corpus and his parser's accuracy was significantly higher than any previously reported, using any of the measures described above. Magerman's algorithm was, however, very complex and it was difficult to investigate the linguistics of the technique, since most of the workings were embedded in the decision tree algorithms.

Expanding on the surprising success of Magerman, Collins (1996) developed a corpus-based algorithm that achieved a parsing accuracy equaling or exceeding Magerman's results, yet was significantly simpler and easier to understand. Collins' approach offers a probabilistic parser that utilizes essential lexical information to model head-modifier relations between pairs of words.

Collins' success extended, in several ways, Magerman's linguistically-grounded insights. The crux of the approach is to reduce every parse tree to a set of (non-recursive) *base noun phrases* and corresponding dependency relationships. For these dependencies, all words internal to a base NP can be ignored, except for the head. Dependencies thus hold between base NP headwords and words in other kinds of constituents. The headword for each phrase is determined from a simple manually-constructed table while the dependency probabilities are estimated from a training corpus. The parsing algorithm itself is a bottom-up chart parser that uses dynamic programming to search the space of all dependencies seen in the training data. In our example sentence, there are five base NPs, as indicated by the following bracketing:

```
[NP John Smith] , [NP the president] of [NP IBM] , announced
[NP his resignation] [NP yesterday] .
```

The Treebank contains enough information to allow an approximation of a version annotated just with base NPs to be constructed automatically. From this, a simple statistical model is automatically constructed that is used to label new material with base NP bracketings. As noted above, each base NP is then reduced to its head for purposes of determining dependency probabilities between pairs of words in the sentence (punctuation is also ignored):

```
Smith president of IBM announced resignation yesterday
```

Dependent words may be either a modifier or an argument of the word it depends on; no distinctions are made among these dependencies here. Each dependency relationship is typed by the three non-terminal labels of the constituents involved in the dependency: the head constituent, the dependent, and the matrix or parent constituent. In our example sentence, the following six dependencies exist:

head	dependent	head label	dependent label	parent label
announced	Smith	<VP	NP	S>
Smith	president	<NP	NP	NP>
president	of	<NP	PP	NP>
of	IBM	<IN	NP	PP>
announced	resignation	<VBD	NP	VP>
announced	yesterday	<VBD	NP	VP>

Given this syntactic model, which is similar in many ways to dependency grammars, and link grammar in particular (Lafferty *et al.* 1992), a parse is simply a set of such dependencies, as well as a set of base NPs. For each new sentence to be parsed, the most likely base NP bracketing is first determined, and then the parser estimates the likelihood of various sets of dependencies (parses), based on the probabilities gleaned from the training corpus. The most likely set of dependencies constitutes the parser's best guess as to the constituent structure of the sentence. The bracketing due to the base NPs is placed on the sentence, and then a labeled bracket can be mapped from each dependency[10] (for example, the last dependency listed above corresponds to the constituent `[VP announced his resignation yesterday]`). After this is done for every sentence in a test corpus, the result can be compared to a key, e.g., the Treebank, and metrics such as those described above can be computed.

The results reported by Collins show the power of such a simple parsing approach. On the *Wall Street Journal* portion of the Treebank, both labeled recall and precision were consistently greater than 84%, matching or bettering Magerman's results in all experiments. The average crossing brackets per sentence was less than 1.5, while between 55 and 60% of the test sentences had no crossing brackets at all, i.e., the constituent structure was completely correct on these sentences (although the labels on the constituents may have differed from the key). Notably, Collins' algorithm is significantly faster than Magerman's; it can parse over 200 sentences per minute, while Magerman's parsing algorithm could parse fewer than ten per minute.[11]

Both Magerman's and Collins' algorithms represented significant breakthroughs in parsing, and it is clear that these breakthroughs could not have taken place without large, annotated corpora such as the Treebank, as well as well-defined evaluation metrics. Nonetheless, it is equally clear that substantial linguistic insight was necessary in order to make good use of the information contained in the corpora.

7. Benefits

As we pointed out when we began, the motivation for adopting a good part of this methodology is that progress can be measured, in very broad and consistent terms. In this section, we review our two major themes with progress in mind.

7.1. The evaluation metric

One of the stated goals of theoretical linguistics has been to develop a complete grammar for a given language; the classic transformational grammar of English compiled by Stockwell, Schacter and Partee (1973) was an attempt to approach just this goal. But a number of difficult problems present themselves almost immediately when we examine such a goal. The first is that although we may have a sense that progress is being made, without some stable paradigm of evaluation we cannot measure our progress toward our goal. No such paradigm has been proposed in theoretical linguistics, as far as we know.

The other problems manifest themselves as soon as we try to define an evaluation metric which is consistent with current theory. There is far more to reaching our goal than simply writing down all the rules a grammar requires. The reason is that any such reasonably large rule set turns out to induce massive ambiguity. In this situation, measuring how close we've come to our goal becomes quite complex. For the sake of simplicity, let us consider only the evaluation of the syntactic component, as outlined in Section 6 above. Instead of the strategies described there, let us assume that our goal is to evaluate any of the many syntactic

theories currently being developed in theoretical linguistics. If this theory permits ambiguity, then we must address this fact in choosing our evaluation metric. One candidate might be that the analysis provided by the key must be one of the analyses permitted by the grammar. But this metric is far too weak; if one assumes a binary branching structure, as is common in linguistic theories, one's grammar could simply generate all possible labelings for all possible binary branchings of any given input and be judged perfect by the evaluation metric! This argument shows that the evaluation metric must be far more strict; in order to have any power, it must demand that the search space of analyses presented to it be narrowed in some substantial way, perhaps even to a single analysis. In other words, providing a set of rules is not enough; the means for choosing between the resulting analyses (that is, a disambiguation strategy) is required as well. Thus the appropriate evaluation metric for theoretical linguistics is how close the grammar and disambiguation strategy come to generating the most appropriate analysis, just as we have shown for CL.

7.2. Confronting the discontinuities

As we've seen, picking the right fundamental linguistic insights is crucial to this paradigm. The part-of-speech tag set used by the Penn Treebank is a distillation of the crucial lexical syntactic distinctions of English; Magerman (1994) and Collins (1996) exploit the notion of syntactic head to derive their syntactic bracketings; Yarowsky (1995) relies on the insight that word senses do not commonly shift within a single discourse to improve his word sense disambiguation algorithm; and Berger *et al.* (1994) identify sublanguages such as names and numbers, perform morphological analysis, and apply syntactic transformations in the course of their statistically-driven translation procedure. But eventually, the benefits of these initial insights are exhausted, and a noticeable error term still remains. In these cases, more expensive, less general insights must be brought to bear; these are the points of discontinuity we've emphasized throughout this article.

For instance, we can determine many syntactic bracketings based simply on part of speech, but additional accuracy can be gained only by referring to lexical subcategorization or semantic class. A good example is PP attachment. PP attachment is no less a problem for current CL than it has been for linguists throughout the ages; in any given sequence of [V N PP], the syntactic key provides an attachment, and the score assigned to our analyses (for example, in terms of crossing bracket measures) is dictated by how closely we conform to the attachments the key provides. This problem is a classic example, of course, of a situation where syntactic information is not particularly helpful. Although the subcategorization frame of the V in question may require a PP and thus provide input to the attachment algorithm, it provides no help when the PP turns out to be a modifier; that is, we cannot distinguish strictly on the basis of subcategorization frames or part-of-speech sequences whether a PP modifier modifies the N or the V. If we need semantic disambiguation, we need to model semantic information in our new paradigm. At this point, one of the crucial differences between the human linguistic understanding task and the computational task manifests itself. In particular, the computer does not have access to the same sorts of semantic generalizations that humans do. In part, what we need to make PP attachment decisions is a ***domain model*****:** of the knowledge of what objects there are in the world, how they can interact with each other, and how likely, prototypical or frequent these interactions are. Humans acquire this information through many sources; in some cases, they read or hear the information, but in most cases (most likely), they acquire this information through direct experience and through the senses.[12] Needless to say, computers do not have access to these data sources, and as a result are at a tremendous disadvantage in semantic tasks. In effect, in attempting semantic analyses

in the corpus-based paradigm, we are forced to imagine how a processor might approach such a task if its only source of information was what it reads.

It turns out that this problem is actually tractable under certain circumstances, as shown by Hindle and Rooth (1993). Their account assumes access to a suitably large set of reasonably correct bracketings, as produced by an algorithm verified by good performance against a bracketed key. This bracketing is *incomplete*; that is, the annotation procedure does not produce constituent structure annotations which it is not reasonably certain of. In terms of our evaluation metrics, the algorithm favors bracketing precision over bracketing recall. Hindle and Rooth take the head relationships corresponding to known instances of PP attachment and use those statistical distributions to predict the unknown cases.[13] In this approach, Hindle and Rooth use lexical heads as an approximation for semantic classes. This approximation is known to be unreliable, because of lexical sense ambiguity; and so others in the field have tackled this problem as well. Yarowsky (1995), for instance, provides a corpus-based algorithm for distinguishing between word senses, based on lists of senses provided from any of a number of sources, including machine-readable dictionaries and thesauri.

None of these analyses are perfect; in fact, some of them perform quite unacceptably in absolute terms. Yet at every step, the limits of simpler approaches are recognized, and the problem is analyzed in terms of identifying the next least complex, the next most powerful, the next most general step to take. And in many of the areas we've discussed here, the field has made substantial progress in the relatively short history of the application of this paradigm. It is safe to say, in fact, that the methodology reviewed here is the only methodology presented so far in theoretical or computational linguistics which can claim to provide *quantifiable* measurement of progress in the field.

8. Conclusion

In the preface to a recent influential dissertation, David Magerman wrote, "I would have liked nothing more than to declare in my dissertation that linguistics can be completely replaced by statistical analysis of corpora (Magerman 1994, p. iv)." Magerman's wish hearkens back to other eras of CL research in which some practitioners in the field hoped to divorce themselves from theoretical linguistics. However, the difference between those periods and corpus-based CL today is that this wish is widely regarded as counterproductive; Magerman himself goes on to conclude that "linguistic input is crucial to natural language parsing, but in a way much different than it is currently being used (p. v)." We have attempted to emphasize this point throughout; while the details of current theory may not be relevant to current corpus-based tasks, the fundamental insights of the theoretical program are central. However, as we've also stressed, the demands of corpus analysis pose substantial theoretical challenges, some of which we've explored here: the nature of discontinuity among linguistic phenomena, the requirements of an evaluation metric for grammar coverage. We have only begun to explore these demands in this article, so by way of conclusion, we summarize two of the other substantial theoretical issues which corpus-based CL raises.

8.1. Coverage vs. depth

The goal of producing a complete grammar for a given language in theoretical linguistics has fallen from favor in recent years, perhaps due to its daunting intractability. In its place, researchers have focused on narrow, deep analyses of particular phenomena, in the hope that a range of such studies will elucidate the general nature of language. Whether or not this process will converge on an unskewed theory of language is an open question. Consider an

analogy with geological research. The exhaustive examination of a single core sample cannot hope to document the geological history of the planet; whether the exhaustive examination of a selection of such samples will produce a fair account of that history depends entirely on whether these samples are representative, given our knowledge of surface topology and the general process of geologic change. It is not clear at all to us that we as linguists possess the knowledge to produce an analogous linguistic sample in an informed way.

The demands of corpus analysis imply a very different strategy. If a computational linguist chooses to parse a year's worth of the ***Wall Street Journal***, she doesn't have the luxury of choosing the sentences she wants to examine; she must analyze *all* of them, in as much detail as the task requires and time and computational resources allow. The general strategy induced by such requirements is broad and shallow, rather than narrow and deep, with added complexity where required. The details of the corpus-based approach may not be appealing to theoretical linguists, but its progress is measurable, and the considerations used to craft these strategies are informed by the same fundamental linguistic insights as those that inform theoretical approaches.

8.2. The nature of data

Another important consequence of this paradigm is that we are severely constrained by the form of the data to be analyzed. Our analysis keys are pairs of raw data and analyses, where "raw" is defined differently for each problem to be evaluated. So for speech recognition, our keys are speech waveforms and their linguistic transcriptions; for part-of-speech tagging, our key is a document and its part-of-speech tags, as illustrated above; for the information extraction tasks, the key is a document and its corresponding database entries. These tasks can be chained; so speech recognition feeds part-of-speech tagging, which in turn feeds information extraction.

There are two important observations to make about data constructed in this way. First, in most cases, the key presented to the system obeys the "no negative evidence" restriction frequently attributed to human language acquisition tasks;[14] second, the properties of the raw data present problems frequently overlooked or idealized away in theoretical linguistics. For instance, the problem of sentence and word segmentation is commonly overlooked, but is crucially relevant to the comprehension process, as demonstrated in Section 5 above. These two observations converge with statistical techniques in a recent article in the journal *Science*, which argues that young infants use probabilistic information about syllable distributions to determine word segmentation in speech (Saffran, Aslin and Newport 1996).

On one remaining significant issue, however, we are currently silent. Although we are convinced that the methodology outlined here ought to have a significant impact on linguistic theory, we do not know what form that impact might take. For instance, one of the primary motivations for examining linguistic questions is to test linguistic theories. However, from the corpus-based point of view, the data thus examined are seriously biased. Parasitic gap constructions, quantifier scope ambiguities, or any one of dozens of deeply-studied linguistic phenomena are infrequently represented in randomly-selected large corpora. Focusing on these examples could well constitute an examination of an unnatural subset of the data, and the resulting generalizations might not extend to the corpus as a whole. We are also aware that while "no negative evidence" is a property of language acquisition, it seems not to account for strong grammaticality judgments by adult speakers. Finally, we do not know what a theory which emphasizes broad coverage over deep analysis might look like. There is no *a priori* reason that the corpus-based methodology would not be applicable to fine-grained linguistic analysis (beyond the significantly larger amount of data which would be required to tease

apart the subtleties in question), but the priorities dictated by broad-coverage analysis suggest that these concerns would necessarily be postponed.

In spite of these uncertainties, we believe, as linguists and computational linguists, that the paradigm we've outlined here is fundamental to genuine progress in language understanding. We also believe that it calls into question a number of common assumptions in mainstream linguistic theory, as a consequence of the demands of large corpus analysis. In this article, we've attempted to make the methodology accessible, to motivate its application, and to highlight its successes, with the hope that more linguists will incorporate this point of view into their daily work.

---

[1]Cf. Schank and Riesbeck 1981, for instance.

[22]For a useful short summary of the history of structuralism, see Newmeyer (1986), chapter 1.

[3]For a discussion, see the papers in Waibel and Lee (1990).

[4]For a detailed discussion of an HMM-based part-of-speech tagger, see Cutting *et al.* (1991). For an application of this technique to higher-level language analysis, see Pieraccini and Levin (1995).

[5]Current speech-based telephone directory assistance, for example, uses this technology.

[6]Context independent utterances were chosen because there are the largest number of comparable data points. The error rate decreased steadily for all of the measures shown, by factors ranging from 4-fold to 9-fold in the period June 1990 to January 1995.

[7]For a thorough discussion of the linguistics of English punctuation, see Nunberg (1990).

[8]For some, the inclusion of null elements in the syntactic annotation may not qualify as "theory-neutral".

[9]The observant reader will note that "yesterday" is unbracketed in the example here. This is because the PARSEVAL evaluation metric requires that singleton brackets be removed before scoring.

[10]Note that many dependency grammar formalisms (as well as many syntactic theories) allow for discontinuous dependencies, while Collins' approach does not. Nor does the bracket-based evaluation framework described here allow for discontinuous constituents.

[11]These parse rates were measured on different computer platforms, but it is clear that Collins' parser is at least an order of magnitude faster than Magerman's. Both approaches are substantially faster than a classical chart parser algorithm.

[12]In fact, there is a substantial body of recent work, typified by Lakoff (1987), that claims that a vast segment of human semantic and linguistic competence is directly inspired by such experiences.

[13]Hindle uses this same strategy in an earlier paper (1990) to generate "concept" clusters and selectional restrictions of verbs.

[14]The notable exception is speech recognition, where false starts and other disfluencies are frequently marked in the annotation.