Pursuing the Goal of Language Understanding

Arun Majumdar, John Sowa, John Stewart

VivoMind Intelligence, Inc.

Abstract. No human being can understand every text or dialog in his or her native language, and no one should expect a computer to do so. However, people have a remarkable ability to learn and to extend their understanding without explicit training. Fundamental to human understanding is the ability to learn and use language in social interactions that Wittgenstein called *language games*. Those language games use and extend prelinguistic knowledge learned through perception, action, and social interactions. This article surveys the technology developed for natural language processing and the successes and failures of various attempts. Although many useful applications have been implemented, the original goal of language understanding seems as remote as ever. Fundamental to understanding is the ability to recognize an utterance as a move in a social game and to respond in terms of a mental model of the game, the players, and the environment. Those models use and extend the prelinguistic models learned through perception, action, and social interactions. Secondary uses of language, such as reading a book, are derivative processes that elaborate and extend the mental models originally acquired by interacting with people and the environment. A computer system that relates language to virtual models might mimic some aspects of understanding, but full understanding requires the ability to learn and use new knowledge in social and sensory-motor interactions. These issues are illustrated with an analysis of some NLP systems and a recommended strategy for the future. None of the systems available today can understand language at the level of a child, but with a shift in strategy there is hope of designing more robust and usable systems in the future.

This is a slightly revised and extended version of a paper in the *Proceedings of the 16th ICCS*, edited by P. Eklund and O. Haemmerlé, LNAI 5113, Springer, Berlin, 2008, pp. 21-42.

1. The Goal of Language Understanding

Some early successes of artificial intelligence led to exaggerated expectations. One example was the theorem prover by Hao Wang (1960), which proved the first 378 theorems of the *Principia Mathematica* in 7 minutes — an average of 1.1 seconds per theorem on the IBM 704, a vacuum-tube machine with 144K bytes of storage. Since that speed was much faster than the two brilliant logicians who wrote the book, pioneers in AI thought that simulating human intelligence would be easy. For machine translation, Delavenay (1960) claimed “While a great deal remains to be done, it can be stated without hesitation that the essential has already been accomplished.” Good (1965) predicted “It is more probable than not that, within the twentieth century, an ultraintelligent machine will be built and that it will be the last invention that man need make.” The movie *2001*, which appeared in 1968, featured the HAL 9000, an intelligent computer that could carry on a conversation in flawless English and even read lips when the humans were trying to communicate in secret. Marvin Minsky, a technical advisor on that movie, claimed it was a “conservative” estimate of AI technology at the end of the 20th century. Yet mathematical tasks, such as proving theorems or playing chess, turned out to be far easier to process by computer than simulating the language skills of a three-year-old child.

In chess or mathematics, a computer can exceed human abilities without simulating human thought. But language is so intimately tied to thought that a computer probably cannot understand language without simulating human thinking at some level. That point raises many serious questions: At what level? With what theory of thinking? With what kinds of internal mechanisms? And with what theories and mechanisms for relating the internal processes via the sensory-motor systems to other agents and the world? Several kinds of theories have been proposed, analyzed, and discussed since antiquity: thoughts are images, thoughts are feelings, thoughts are propositions, and thoughts are multimodal combinations of images, feelings, and propositions.

The propositional theory has been the most popular in AI, partly because it's compatible with a large body of work in logic and partly because it's the easiest to implement on a digital computer. Figure 1 illustrates the classical paradigm for natural language processing. At the top is a lexicon that maps the vocabulary to speech sounds, word forms, grammar, and word senses. The arrows from left to right link each stage of processing: phonology maps the speech sounds to phonemes; morphology relates the phonemes to meaningful units or morphemes; syntax analyzes a string of morphemes according to grammar rules; and semantics interprets the grammatical patterns to generate propositions stated in some version of logic.

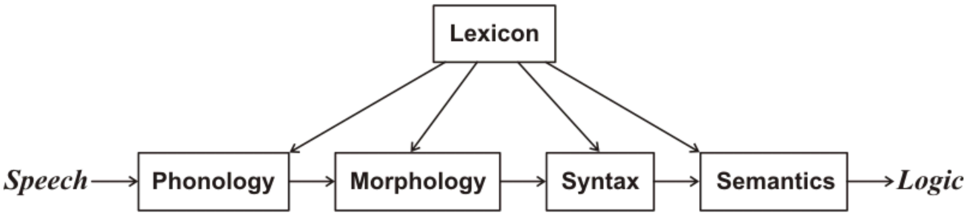


Figure 1. Classical stages in natural language processing

Psycholinguistic evidence since the 1960s has shown that Figure 1 is unrealistic. All the one-way arrows should be double headed, because feedback from later stages has a major influence on processing at earlier stages. Even the arrows from the lexicon should be double headed, because people are constantly learning and coining new words, new word senses, and new variations in syntax and pronunciation. The output labeled *logic* is also unrealistic, because logicians have not reached a consensus on an ideal logical form and many linguists doubt that logic is an ideal representation for semantics. Furthermore, Figure 1 omits everything about how language is used by people who interact with each other and the world. Figure 2 is a more realistic diagram of the interconnections among the modules.

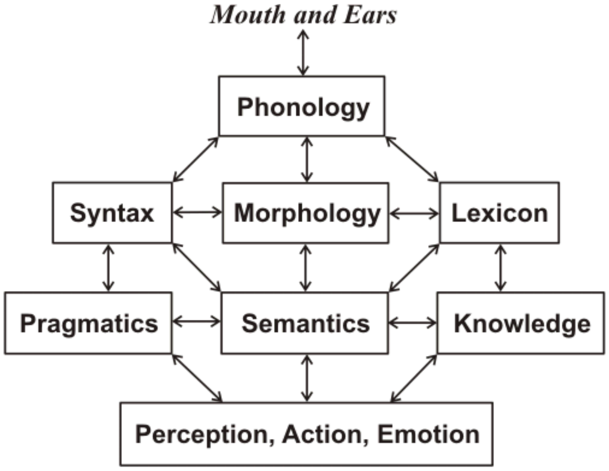


Figure 2. A more realistic diagram of interconnections

Yet Figure 2 also embodies questionable assumptions. The box labeled *perception, action, and emotion*, for example, blurs all the levels of cognition from fish to chimpanzees. Furthermore, the boxes of

Figure 2 correspond to traditional academic fields, but there is no evidence that those fields have a one-to-one mapping to modules for processing language in the brain. In particular, the box labeled *knowledge* should be subdivided in at least three ways: language-independent knowledge stored in image-like form; conceptual knowledge related to language, but independent of any specific language; and knowledge of the phonology, vocabulary, and syntax of specific languages. The box labeled *pragmatics* deals with the use of language in human activities. Wittgenstein (1953) proposed a reorganization in *language games*, according to the open-ended variety of ways language is used in social interactions. That subdivision would cause a similar partitioning of the other boxes, especially semantics, knowledge, and the lexicon. It would also affect the variations of syntax and phonology in casual speech, professional jargon, or "baby talk" with an infant.

In his first book, Wittgenstein (1921) presented a theory of language and logic based on principles proposed by his mentors, Frege and Russell. Believing he had solved all the problems of philosophy, Wittgenstein retired to an Austrian mountain village, where he taught elementary schoolchildren. Unfortunately, the children did not learn, think, or speak according to those principles. In his second book, Wittgenstein (1953) systematically analyzed the "grave errors" (*schwere Irrtümer*) in the framework he had adopted. One of the worst was the view that logic is superior to natural languages and should replace them for scientific purposes. Frege (1879), for example, hoped "to break the domination of the word over the human spirit by laying bare the misconceptions that through the use of language often almost unavoidably arise concerning the relations between concepts." Russell shared Frege's low opinion of natural language, and both of them inspired Carnap, the Vienna Circle, and most of analytic philosophy.

Many linguists and logicians who work within the paradigm of Figure 1 admit that it's oversimplified, but they claim that simplification is necessary to enable researchers to address solvable subproblems. Yet Richard Montague and his followers have spent forty years working in that paradigm, and computational linguists have been working on it for half a century. But the goal of designing a system at the level of HAL 9000 seems more remote today than in 1968. Even pioneers in the logic-based approach have begun to doubt its adequacy. Kamp (2001), for example, claimed "that the basic concepts of linguistics — and especially those of semantics — have to be thought through anew" and "that many more distinctions have to be drawn than are dreamt of in current semantic theory."

This article emphasizes the distinctions that were dreamt of and developed by cognitive scientists who corrected or rejected the assumptions by Frege, Russell, and their followers. Section 2 begins with the semeiotic by Charles Sanders Peirce, who had invented the algebraic notation for logic, but who placed it in a broader framework than the 20th-century logicians who used it. Section 3 discusses the ubiquitous pattern matching in every aspect of cognition and its use in logical and analogical reasoning. Section 4 presents Wittgenstein's language games and the social interactions in which language is learned, used, and understood. Section 5 introduces Minsky's *Society of Mind* as a method of supporting the interactions illustrated in Figure 2. Section 6 summarizes the lessons learned from work with two earlier language processors. The concluding Section 7 outlines a multilevel approach to language processing that can support more robust and flexible systems.

2. Semeiotic and Biosemiotics

Peirce claimed that the primary characteristic of life is the ability to recognize, interpret, and respond to signs. Signs are even more fundamental than neurons because every neuron is itself a semiotic system: it receives signs and interprets them by generating more signs, which it passes to other neurons or muscle cells. Every cell, even an independent bacterium, is a semiotic system that recognizes chemical, electrical, or tactile signs and interprets them by generating other signs. Those signs can cause the walls of a bacterial cell to contract or expand and move the cell toward nutrients and away from toxins. The brain is a large colony of neural cells, whose signs coordinate a symbiotic relationship within an organism of many kinds of cells. The neural system supports rapid, long-distance communication by electrical signs, but all cells can communicate locally by chemical signs. By secreting chemicals into the blood stream, cells can broadcast signs by a slower, but more pervasive method. At every level from a single cell to a multicellular organism to a society of organisms, signs support and direct all vital processes. *Semeiotic* is Peirce's term for the theory of signs. The modern term *biosemiotics* emphasizes Peirce's point that sign processing is more general than human language and cognition.

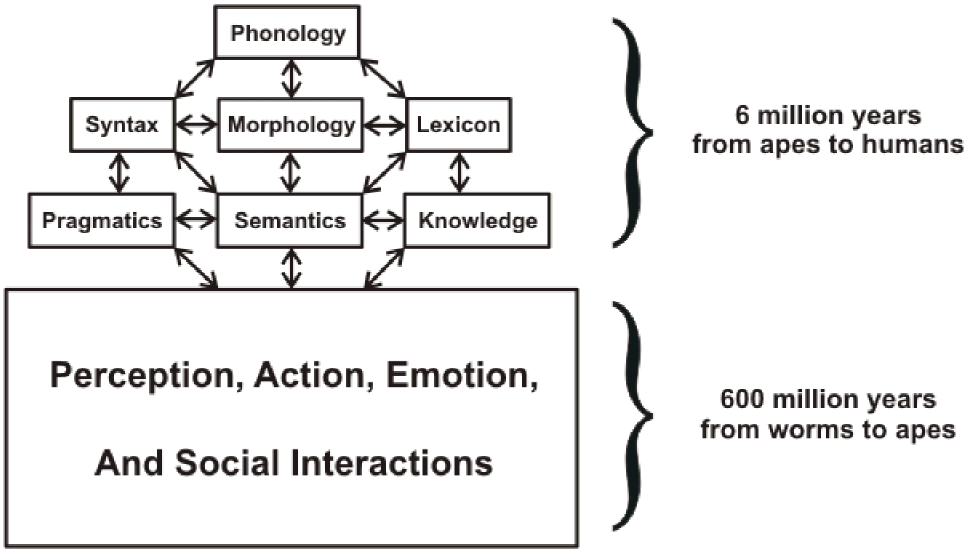

Phonology
Syntax
Morphology
Lexicon
Pragmatics
Semantics
Knowledge
6 million years from apes to humans
Perception, Action, Emotion, And Social Interactions
600 million years from worms to apes

Figure 3. An evolutionary view of the language modules

Deacon (1997), a professional neuroscientist, used Peirce's theories as a guide for relating neurons to language. Figure 3 illustrates his view that the language modules of the brain are a recent addition and extension of a much older ape-like architecture. Deacon used Peirce's categories of *icon*, *index*, and *symbol* to analyze the signs that animals recognize or produce. The calls a hunter utters to control the dogs are indexes, the vocal equivalent of a pointing finger. Vervet monkeys have three types of warning calls: one for eagles, another for snakes, and a third for leopards. Some people suggested that those calls are symbols of different types of animals, but vervets never use them in the absence of the stimulus. More likely, the vervet that sees the stimulus uses the call as an index to tell other vervets to look up, look down, or look around. An early step from index to symbol probably occurred when some hominin proposed a hunt by uttering an index for prey, even before the prey was present. After symbols became common, they would enable planning and organized activities in every aspect of life. The result would be a rapid increase in vocabulary, which would promote the *co-evolution* of language, brain, vocal tract, and culture.

Like Frege, Peirce was a logician who independently developed a complete notation for first-order logic. Unlike Frege, Peirce had a high regard for the power and flexibility of language, and he had worked as an associate editor of the *Century Dictionary*, for which he wrote, revised, or reviewed over 16,000 definitions. Peirce never rejected language or logic, but he situated both within the broader theory of signs. In his semeiotic, every sign is a triad that relates a perceptible *mark* (1), to another sign called its *interpretant* (2), which determines an existing or intended *object* (3). Following is one of Peirce's most often quoted definitions:

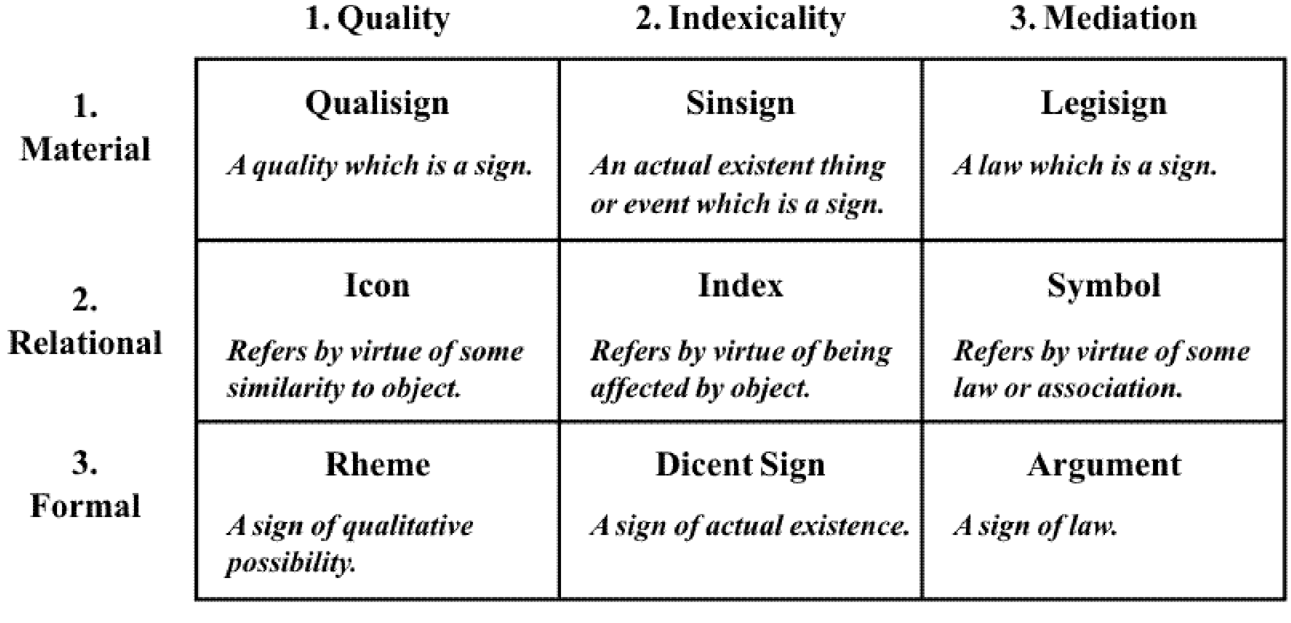
> A sign, or *representamen*, is something which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an

> equivalent sign, or perhaps a more developed sign. That sign which it creates I call the *interpretant* of the first sign. The sign stands for something, its *object*. It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the *ground* of the representamen. (CP 2.228)

A pattern of green and yellow in the lawn, for example, is a mark, and the interpretant is some type, such as Plant, Weed, Flower, SaladGreen, or Dandelion. The guiding idea that determines the interpretant depends on the context and the intentions of the observer. The interpretant determines the word the observer chooses to express the experience. The listener who hears that word uses background knowledge to derive an equivalent interpretant.

As Peirce noted, an expert with a richer background can sometimes derive a more developed interpretant than the original observer. Mohanty (1982:58) remarked “Not unlike Frege, Husserl would rather eliminate such fluctuations from scientific discourse, but both are forced to recognize their recalcitrant character for their theories and indispensability for natural languages.” Communication in which both sides have identical interpretants is possible with computer systems. Formal languages are precise, but they are rigid and fragile. The slightest error can and frequently does cause a total breakdown, such as the notorious “blue screen of death.”

On the surface, Peirce’s triads seem similar to the *meaning triangles* by Aristotle, Frege, or Ogden and Richards (1923). The crucial difference is that Peirce analyzed the underlying relationships among the vertices and sides of the triangle. By analyzing the relation between the mark and its object, Peirce (1867) derived the triad of *icon, index* and *symbol*: an icon refers by some similarity to the object; an index refers by a physical effect or connection; and a symbol refers by a law, habit, or convention. Figure 4 shows this *relational* triad in the middle row.



	1. Quality	2. Indexicality	3. Mediation
1. Material	**Qualisign** *A quality which is a sign.*	**Sinsign** *An actual existent thing or event which is a sign.*	**Legisign** *A law which is a sign.*
2. Relational	**Icon** *Refers by virtue of some similarity to object.*	**Index** *Refers by virtue of being affected by object.*	**Symbol** *Refers by virtue of some law or association.*
3. Formal	**Rheme** *A sign of qualitative possibility.*	**Dicent Sign** *A sign of actual existence.*	**Argument** *A sign of law.*

Figure 4. Peirce’s triple trichotomy

Later, Peirce added the first row or *material triad*, which signifies by the nature of the mark itself. The third row or *formal triad* signifies by a formal rule that relates all three vertices — the mark, interpretant, and object. The basic units of language are characterized by the formal triad: a word serves as a rheme; a sentence, as a dicent sign; and a paragraph or other sequence, as an argument. The labels at the top of Figure 4 indicate how the sign directs attention to the object: by some quality of the mark, by some causal or pointing effect, or by some mediating law, habit, or convention. The following examples illustrate nine types of signs:

1. **Qualisign** (material quality). A ringing sound as an uninterpreted sensation.
2. **Sinsign** (material indexicality). A ringing sound that is recognized as coming from a telephone.
3. **Legisign** (material mediation). The convention that a ringing telephone means someone is trying to call.
4. **Icon** (relational quality). An image that resembles a telephone when used to indicate a telephone.
5. **Index** (relational indexicality). A finger pointing toward a telephone.
6. **Symbol** (relational mediation). A ringing sound on the radio that is used to suggest a telephone call.
7. **Rheme** (formal quality). A word, such as *telephone*, which can represent any telephone, real or imagined.
8. **Dicent Sign** (formal indexicality). A sentence that asserts an actual existence of some object or event: "You have a phone call from your mother."
9. **Argument** (formal mediation). A sequence of dicent signs that expresses a lawlike connection: "It may be an emergency. Therefore, you should answer the phone."

The nine categories in Figure 4 are more finely differentiated than most definitions of signs, and they cover a broader range of phenomena. Anything that exists can be a sign of itself (sinsign), if it is interpreted by an observer. But Peirce (1911:33) did not limit his definition to human minds or even to signs that exist in our universe:

> A sign, then, is anything whatsoever — whether an Actual or a May-be or a Would-be — which affects a mind, its Interpreter, and draws that interpreter's attention to some Object (whether Actual, May-be, or Would-be) which *has already* come within the sphere of his experience.

The mind or quasi-mind that interprets a sign need not be human. In various examples, Peirce mentioned dogs, parrots, and bees. Higher animals typically recognize icons and indexes, and some might recognize symbols. A language of some kind is a prerequisite for signs at the formal level of rhemes, dicent signs, and arguments.

As these examples show, Peirce's theory of signs provides a more nuanced basis for analysis than the all-or-nothing question of whether animals have language. Unlike the static meaning triangles of Aristotle or Frege, the most important aspect of Peirce's triangles is their dynamic nature: any vertex can spawn another triad to show three different perspectives on the entity represented by that vertex. During the course of a conversation, the motives of the participants lead the thread of themes and topics from triangle to triangle.

3. Perception, Cognition, and Reasoning

Language affects and is affected by every aspect of cognition. Only one topic is more pervasive than language: signs in general. Every cell of every organism is a semiotic system, which receives signs from the environment, including other cells, and interprets them by generating more signs, both to control its own inner workings and to communicate with other cells of the same organism or different organisms. The brain is a large colony of neural cells, which receives, generates, and transmits signs to other cells of the organism, which is an even larger colony. Every publication in neuroscience describes brains and neurons as systems that receive signs, process signs, and generate signs. Every attempt to understand those signs relates them to other signs from the environment, to signs generated by the organism, and to theories of those signs in other branches of cognitive science. The meaning of the neural signs can only be determined by situating neuroscience within a more complete theory that encompasses every aspect of cognitive science.

By Peirce's definition of *sign*, all life processes, especially cognition, involve receiving, interpreting, generating, storing, and transmitting signs and patterns of signs. Experimental evidence is necessary to determine the nature of the signs and the kinds of patterns generated by the interpretation. Perceptual signs are *icons* derived from sensory stimulation caused by the outside world or caused by internal bodily processes. Recognition consists of interpreting a newly received icon by matching it to previously classified icons called *percepts* and patterns of percepts called *Gestalts*. The interpretation of an icon is the pattern formed by the percepts, Gestalts, and other associated signs. The interpreting signs may be image-like percepts or imageless *concepts*, which are similar to percepts, but without the sensory connections.

Analogy is a method of reasoning based on pattern matching, and every method of logic is a constrained use of analogy. As an example, consider the rule of deduction called *modus ponens*:

```
Premise:     If P then Q.
Assertion:   P′.
Conclusion:  Q′.
```

This rule depends on the most basic form of pattern matching: a *comparison* of P and P′ to determine whether they are identical. If P in the premise is not identical to P′ in the assertion, then a pattern-matching process called *unification* specializes P by some transformation S that makes S(P) identical to P′. By applying the same specialization S to Q, the conclusion Q′ is derived as S(Q). Each of the following three methods of logic constrain the pattern matching to specialization, generalization, or identity.

1. **Deduction.** Specialize a general principle.

```
Known:  Every bird flies.
Given:  Tweety is a bird.
Infer:  Tweety flies.
```

2. **Induction.** Generalize multiple special cases:

```
Given:  Tweety is a bird. Polly is a bird. Hooty is a bird.
        Tweety flies. Polly flies. Hooty flies.
Assume: Every bird flies.
```

3. **Abduction.** Given a special case and a known generalization, make a guess that explains the special case.

```
Given:  Tweety flies.
Known:  Every bird flies.
Guess:  Tweety is a bird.
```

These three methods of logic depend on the ability to use symbols. In deduction, the general term *every bird* is replaced by the name of a specific bird *Tweety.* Induction generalizes a property of multiple individuals — Tweety, Polly, and Hooty — to the category Bird, which subsumes all the instances. Abduction guesses the new proposition *Tweety is a bird* to explain one or more observations. According to Deacon's hypothesis that symbols are uniquely human, these three reasoning methods could not be used by nonhuman mammals.

According to Peirce (1902), “Besides these three types of reasoning there is a fourth, analogy, which combines the characters of the three, yet cannot be adequately represented as composite.” Its only prerequisite is *stimulus generalization* — the ability to classify similar patterns of stimuli as signs of similar objects or events. Unlike the more constrained operations of generalization and specialization, similarity may involve a generalization of one part and a specialization of another part of the same pattern. Analogy is more primitive than logic because it does not require language or symbols. In Peirce’s terms, logic requires symbols, but analogy can also be performed on image-like icons. Case-based reasoning (CBR) is an informal method of reasoning, which uses analogy to find and compare cases that may be relevant to a given problem or question.

Whether the medium consists of discrete words or continuous images, CBR methods start with a question or goal Q about some current problem or situation P. By analogy, cases that resemble P are recalled from long-term memory and ranked according to their similarity to P. The case with the greatest similarity (i.e., smallest semantic distance) is the most likely to answer the question Q. When a similar case is found, the part of the case that matches Q is the predicted answer. If two or more cases are similar to P, they might not predict the same answer. If they do, that answer can be accepted with a high degree of confidence. If not, multiple cases can be combined by some transformation: a disjunction (Q1 or Q2), a generalization of Q1 and Q2, or a blend of features from both. A semantic distance measure could be used to choose the most appropriate transformation by comparing the results with typical examples in the knowledge base.

Both logic and CBR have a large overlap on which they’re compatible: they would generate consistent responses to the same questions. For highly regular data, induction can generalize many cases to rules of the form *If P, then Q*. For such data, CBR would derive the same conclusions as a method of deduction called *backward chaining*: a goal Q′ is unified to the conclusion Q of some if-then rule by means of a specialization S; the application of S to P produces the pattern P′, which is a generalization of one or more cases. Formal deduction is best suited to thoroughly analyzed areas of science, for which induction can reduce a large number of cases to a small number of rules. CBR is most valuable for subjects with highly varied or frequently changing cases, for which any axioms would have a long list of exceptions. In legal reasoning, for example, the list of laws and the list of cases are enormous, and nearly every generalization has as many exceptions as applications.

For both formal and informal reasoning, a high-speed method of indexing and finding relevant data is essential, but discrete list-processing methods have been too slow. The world is continuous, all physical motions are continuous, feelings and sensations vary continuously, but every natural language has a discrete, finite set of meaningful units or *morphemes*. No discrete set of symbols can faithfully represent a continuous world, but a cognitive system must map discrete words to and from continuous sensations. Wildgen (1982, 1994) maintained that continuous fields are the primary basis for perception and cognition, and he adopted René Thom’s *catastrophe-theoretic semantics* for identifying the patterns that map to the discrete words and phrases. That approach is still controversial, but the principle of mapping discrete structures such as *conceptual graphs* (CGs) to continuous fields has proved to be valuable for developing efficient methods for indexing CGs and computing the semantic distance between them (Sowa & Majumdar 2003). Those methods were used for finding analogies by the VivoMind Analogy Engine (VAE), and more precise and flexible mappings have been implemented in a new system called Cognitive Memory™. This system is based on active agents, as discussed in Section 5, and it encodes arbitrarily large conceptual graphs in Cognitive Signatures™, which are mathematical structures embedded in a continuous field. Psychologically, those signatures represent *chunks* of knowledge that can be related to other chunks by high-speed numeric computations.

4. Language Games

The first language game may have evolved about four million years ago as a system of grunts and gestures for organizing a hunt. At that time, chimpanzees lived in the forests of west Africa, while our ancestor, *Australopithecus*, lived in the grasslands to the east. With fewer trees to climb, the Australopithecines began to walk upright. Chimps supplement their diet by catching and eating small game, but in lands with sparser vegetation, Australopithecines required more meat. Since they weren't as fast as their prey, they had to hunt in organized parties, which require communication. The calls and gestures of the chimps were adequate for occasional hunting, but when hunting became a necessity, any improvement in communication would be an enormous advantage. Vocal calls are convenient because they leave the hands free, and they can be spoken and heard while eyes are focused on the prey. The earliest protowords were probably a few dozen indexical signs, of the sort that modern hunters and shepherds use to control their dogs. The first step from index to symbol likely occurred when some hominin proposed a hunt by uttering the index for prey, even before the prey was present. After symbols were invented, language games could be integrated with every social activity that involved cooperation, negotiation, persuasion, planning, or play.

Wittgenstein's theory of language games has major implications for both semantic theory and computational linguistics. It implies that the ambiguities of natural language are not the result of careless speech by uneducated people. Instead, they result from the fundamental nature of language and the way it relates to the world: each language uses and reuses a finite number of words to represent an unlimited number of topics. A closed semantic basis along classical lines is not possible for any natural language. Instead of assigning a single meaning or even a fixed set of meanings to each word, a theory of semantics must permit an open-ended number of meanings:

- Words are like playing pieces that may be used and reused in different language games.
- Associated with each word is a limited number of lexical patterns that are common to all the language games that use the word.
- Meanings are deeper conceptual patterns that change from one language game to another.
- Metaphor and conceptual refinement are techniques for transferring the lexical patterns of a word to a new language game and thereby creating new conceptual patterns for that game.

Once a lexical pattern is established for a concrete domain, it can be transferred by metaphor to create similar patterns in more abstract domains. By this process, an initial set of lexical patterns can be built up; later, they can be generalized and extended to form new conceptual patterns for more abstract subjects. The possibility of transferring patterns from one domain to another increases flexibility, but it leads to an inevitable increase in ambiguity.

If the world were simpler, less varied, and less changeable, natural languages might be unambiguous. But the complexity of the world causes the meanings of words to shift subtly from one domain to the next. If a word is used in widely different domains, its multiple meanings may have little or nothing in common. As an example, the word *invest*, which originally meant to put on clothing, has come to mean either to surround a fortress or to make a certain kind of financial transaction. In Italian, the related word *investmento* has all the senses of the English *investment*, but with the added sense of traffic accident. As these examples illustrate, the mechanisms of natural languages not only permit, but actually facilitate arbitrarily large shifts in meaning. They have enabled isolated tribes using stone-age tools to adapt to 21st-century cultures within the lifetime of a single generation, while continuing to speak what is called "the same language."

Although Wittgenstein's theory of language games has been highly influential, some linguists and philosophers have raised criticisms and proposed alternative, but related hypotheses. Hattiangadi (1987) proposed that the meaning of a word is the set of all possible *theories* in which it may be used, but that term sounds too formal to cover everyday speech. Kittredge and Lehrberger (1982) used the term *sublanguage* for any specialized language used in any context for any purpose. Whatever it's called, a language game or any related variation must involve an organized system of language patterns and practices that are intimately bound to a system of behavior — or, as Wittgenstein called it, a way of life. Language can only be understood in terms of the social activities of its speakers. Full understanding of the language would require a person or robot to participate in the activity in a way that other participants would consider appropriate. This requirement, which is a variant of the Turing test, is a necessary condition for a single language game. A sufficient condition for general understanding would require the ability to learn, use, and invent a wide range of language games in the same kinds of circumstances as native speakers.

By those criteria, the bonobo Kanzi is a nonhuman person who has reached a level of language understanding that is beyond the ability of any computer system yet devised (Savage-Rumbaugh & Lewin 1994). On a test of spoken English with sentences such as "Get the rubber band that's in the bathroom," Kanzi responded with the correct action to 72% of the sentences; Alia, a two-year-old girl, responded correctly to 66% of them. Even more impressive are the reports by Stuart Shanker, a skeptical Wittgensteinian philosopher who became a believer after visiting Kanzi and his teacher, Sue Savage-Rumbaugh. On Shanker's first visit, Sue asked Kanzi, "I'm going to take Stuart around the lab. Could you please water the tomato plants for me while we're doing this?" Following is Shanker's description of what Kanzi did:

> And sure enough, I watched as he trundled over to an outdoor water faucet, picked up a bucket that was lying beside it, turned on the spigot and filled the bucket, turned off the faucet himself, and then walked down to a vegetable patch at the far end of the compound, carrying the bucket in one hand. When he reached the vegetables, I watched as he poured the water on a small patch of tomato plants growing in the corner of the vegetable patch. (Greenspan & Shanker 2004:107).

There is no evidence of which words of the request Kanzi understood. But he was undoubtedly familiar with the task of watering the tomatoes, and he understood the language games related to that task. Linguists have claimed that the inability to produce detailed syntax indicates that apes have not learned a truly natural language. Yet apes and two-year-old children satisfy the criteria of learning, using, and inventing language games integrated with their behavior. If Kanzi were human, he would be diagnosed as having a language deficit, but no one would doubt his understanding of the language associated with the activities in which he participated.

Current natural language processors have been used in many valuable applications, such as translating languages, finding and extracting information from text, summarizing texts, answering questions, and checking and correcting syntax errors. Some of them have been used to control robots, but none of them have been able to learn, play, and invent language games at the level of Kanzi and other apes. Furthermore, none of them have been able to learn, use, and generate language at the level of a three-year-old child. The following sentences were uttered by a child named Laura at 2 years, 10 months (Limber 1973):

> *Here's a seat. It must be mine if it's a little one.*
>
> *I went to the aquarium and saw the fish.*
>
> *I want this doll because she's big.*

When I was a little girl, I could go "geek geek" like that,
but now I can go "This is a chair."

Forty years ago, the goal of AI was to meet or exceed all human intellectual abilities. Today, reaching the level of Laura or Kanzi would be a major achievement.

Laura's sentences contain implications (*if*), causal connections (*because*), modal auxiliaries (*can* and *must*), contrast between past and present tenses, metalanguage about her own language at different times, and parallel stylistic structure. Combining modal, temporal, causal, and metalevel logic and reasoning in a single formalism and using it to interpret and generate natural language is still a major research topic. Even though she couldn't prove theorems as fast as Wang's program, Laura used all those operators before the age of three. The assumption that formal logic is the foundation or prerequisite for language understanding seems unlikely.

Some linguists and philosophers have been searching for an elusive "natural logic" that underlies language. Yet there is no sharp boundary between ordinary language and any formal logic. When two logicians talk on the telephone, they can convey the most abstruse ideas with a few words added to ordinary language. A better assumption is that formal logic is a language game played with symbols and patterns abstracted from natural languages. Formal logic may sound unnatural to the uninitiated, but that is true of the language games of any specialized field. Sailors, plumbers, chefs, and computer hackers scorn the "book learning" of novices who try to use their jargon without mastering the associated skills. Book learning is useful, but computer systems must relate it to action in order to demonstrate understanding at the level of Laura or Kanzi.

Some language games involve a disciplined use of syntax, semantics, and vocabulary in a *controlled natural language* that a computer can process without full understanding. An example is the METEO system for translating weather reports to and from English and French (Thouin 1982). For routine reports about the temperature and precipitation, METEO does the translation without human assistance, but for unusual conditions outside the controlled subset, it sends the reports to a human translator. Speech recognition systems for handling telephone calls frustrate people who need to discuss situations that fall outside the controlled subset. More research is necessary to broaden the controlled subsets and determine when to transfer the call to a human being.

5. Society of Mind

In computer systems, the linear flow of Figure 1 is easy to implement: each stage analyzes some data, passes the results to the next stage, and never sees the same data again. But the more complex interconnections of Figure 2 allow other modules, even later stages, to request or propose different interpretations of previously analyzed data. In recordings of the following sentences, for example, Warren (1970) spliced a patch of white noise at each point marked ¿:

The ¿eel is on the shoe.

The ¿eel is on the car.

The ¿eel is on the table.

The ¿eel is on the orange.

Although the sound was identical in each of the four sentences, the listeners who heard the recordings interpreted the four words as *heel*, *wheel*, *meal*, and *peel*, respectively. Apparently, feedback from the semantic stage caused a reinterpretation of the phonology of an earlier word in the sentence. Furthermore, the listeners were not aware of hearing anything unusual. Many similar studies indicate a great deal of parallel processing in the brain with feedback from later stages to earlier stages, usually at a level beneath conscious awareness. To support parallel processing with feedback, a computer system would require a more complicated control structure than a linear flow.

An early AI model of parallel reasoning was Pandemonium (Selfridge 1959), which consisted of a collection of autonomous agents called *demons*. Each demon could observe aspects of the current situation or workspace, perform some computation, and put its results back into the workspace. In effect, Pandemonium was a parallel *forward-chaining* reasoner, whose major drawback was that the demons generated large volumes of mostly useless data that overflowed available storage. For a more disciplined method of passing messages among the linguistic modules, Hahn et al. (1994) designed ParseTalk as a distributed, concurrent, object-oriented parser. In discussing its advantages, the authors noted that ParseTalk replaces "the static global-control paradigm" of Figure 1 with "a dynamic, local-control model" that supports "a balanced treatment of both declarative and procedural constructs within a single formal framework." Although ParseTalk is a promising approach, Figure 3 suggests that the language modules use a much older and more pervasive system that supports all aspects of perception, cognition, and action. Therefore, the large box at the bottom of Figure 3 should also be subdivided in modules that operate in parallel and communicate by message passing.

The integration of language games with social activity implies that the language modules should be further subdivided and interconnected with other cognitive modules in dynamically changing ways. The modules for reading, for example, would connect visual perception to the syntactic and semantic mechanisms. Psycholinguistic studies with Japanese syllabic kana symbols and character-based kanji, indicates that they use different neural mechanisms even for reading. Singing integrates language and music in ways that make both the words and the melodies easier to recognize and reproduce. Singing is also connected to dancing, marching, and various kinds of rhythmic work and play. Some linguists claimed that music was based on the syntactic mechanisms of language, but Mithen (2006) presented detailed evidence to show that music is older and independent of language. In fact, syntax may have evolved with or from the music of prosody. Whatever the basis, the number of modules is probably far greater than the eight boxes of Figures 2 and 3. Perhaps there is no limit to the number of modules, and every language game and mode of behavior has its own module or even a collection of interacting modules.

The diversity of mechanisms associated with language is a subset of the even greater diversity involved in all aspects of cognition. In his book *The Society of Mind*, Minsky (1987) surveyed that diversity and proposed an organization of active agents as a computational model that could simulate the complexity:

> What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity, not from any single, perfect principle. Our species has evolved many effective although imperfect methods, and each of us individually develops more on our own. Eventually, very few of our actions and decisions come to depend on any single mechanism. Instead, they emerge from conflicts and negotiations among societies of processes that constantly challenge one another. (Section 30.8)

This view is radically different from the assumption of a unified formal logic that cannot tolerate a single inconsistency. Unlike the ParseTalk goal of "a single formal framework," Minsky's goal is to build a flexible, fault-tolerant system out of imperfect, possibly fallible components. Such a system can support logic, just as the flexible, fault tolerant, and fallible human brain supports logic. More recently, Minsky (2006) emphasized the role of emotions in driving an engine composed of multiple agents. Without emotions to set the goals, a logic-based theorem prover would have no reason to do anything.

Minsky's proposal for a society of interacting agents could be implemented in a variety of ways. The Flexible Modular Framework™ (FMF) proposed by Sowa (2002, 2004) is an architecture for intelligent systems that was inspired by Minsky's society of agents, by McCarthy's proposal for the logic-based language Elephant 2000, and by Peirce's semeiotic. As in Minsky's society, each module in the FMF is an autonomous agent that communicates with other agents by passing messages. As in McCarthy's Elephant, each message specifies a speech act that indicates its purpose, and the messages may be expressed in logic. As in Peirce's semeiotic, each message is a sign at any level of complexity, each agent is a "quasi-mind" that interprets signs by generating other signs, and many agents use the Peirce-inspired system of logic called *conceptual graphs*. An agent that knows another agent's identity can send it a message directly, but any agent can post a message to a Linda blackboard, and any other agent that can process that type of message can respond to it (Gelernter 1985). Unlike ParseTalk, the FMF does not require a single formal framework, but it can support the *π-calculus*, which is a generalization of Petri nets that allows new agents and communication paths to be created or destroyed dynamically (Milner 1999). Several variations of the FMF have been implemented, and all of them use a lightweight protocol that can be implemented in 8K bytes per agent. Thousands of agents can run simultaneously on a laptop computer, but they can communicate with other agents anywhere across the Internet.

An interactive system of agents that can change their configuration dynamically is strictly more expressive than a conventional Turing machine, and it can compute functions that are not Turing computable (Eberbach et al. 2004). The π-calculus is one such system. Another is the *$-calculus*, which has the same operators as the π-calculus, but adds a *cost* measure for each computation. A cost measure based on space and time requirements can constrain the excesses of systems like Pandemonium by rewarding agents that produce good results with more resources and reducing the resources of agents that produce useless data.

At VivoMind, the authors have developed a learning method called *Market-Driven Learning* (MDL), which uses a version of $-calculus. The basic idea is that the system of agents is organized in a managerial hierarchy with one agent called the CEO at the top. The CEO is responsible for producing results that earn rewards, measured in units of space and time, in order to keep the society of agents in business. At the bottom of the hierarchy are agents that find data, combine data, or propose hypotheses. Some of them are freelance agents who sell data or hypotheses by advertising them on the Linda blackboards. Other agents are hired by some agent that serves as a manager. Each manager has one or more agents as employees, and every manager except the CEO is allocated resources by a higher-level manager. The managers can use their resources to hire employees, reward employees for good performance, or buy data and hypotheses from freelance agents or from other managers. The managers may combine the data and hypotheses themselves, assign their employees the task of doing the combination, or serve on a committee with other managers to produce a combined report.

The MDL society learns by reorganizing itself to produce consistently good results, which humans are willing to buy. The rewards pass through the hierarchy from manager to employee and create an effect of *backward propagation* similar to the learning methods of a neural network. But unlike the simple switches and numeric functions of a neural network, the MDL agents can be arbitrarily complex programs or reasoning systems, they can hire or fire other agents, and the messages can be propositions or even large documents stated in some version of logic. Also unlike a neural network, the messages that pass through the MDL can be translated to Common Logic Controlled English (CLCE) in order to provide humanly readable explanations about the way any agent derived its data, hypotheses, or reports. By simulating a variety of business methods, the MDL approach has produced good results, and it is used in the VivoMind Language Processor described in Section 7.

6. Experience with Intellitex and CLCE

The theoretical issues discussed in the previous sections influenced the design of two language processors developed and used by the authors: the Intellitex parser, which produced an approximate translation from English to conceptual graphs, and the CLCE parser, which translated the formally defined subset of Common Logic Controlled English to precisely defined conceptual graphs. The two parsers had complementary strengths and weaknesses:

- Intellitex was a fast, but shallow parser that used a version of link grammar (Sleator & Temberley 1993) to translate English sentences to conceptual graphs. Intellitex always generated some conceptual graph as an approximation to the semantics of an English sentence, but its grammar and semantics were not sufficiently detailed to generate an accurate logical form for complex sentences. Its approximations, however, were useful for many applications, such as analogical reasoning and question answering (Sowa & Majumdar 2003). The VivoMind Analogy Engine (VAE) could enhance and correct the approximate CGs generated by Intellitex, but only if a large knowledge base of precisely defined CGs happened to be available. For an important application, such knowledge enabled Intellitex to perform amazingly well.
- The CLCE parser was a traditional syntax-directed parser, which processed character strings written in Common Logic Controlled English (Sowa 2004). It followed the stages from morphology to semantics in Figure 1 to generate a logical form in the Conceptual Graph Interchange Format (CGIF), as defined by the ISO/IEC 24707 standard for Common Logic. But the CLCE subset of English is a formal language, and the CLCE parser was as rigid and unforgiving as a parser for any formal language. Making it more user friendly or extending it to a broader range of English constructions would require a large number of grammar rules. Furthermore, each grammar rule would require a corresponding semantic rule to generate the correct CG.

Over time, incremental improvements were made to both of these processors, but their methods for parsing English and generating CGs were so different that no synergism between them was possible. Intellitex was more robust and forgiving than the CLCE parser, but it could not detect and correct errors in the input that would cause the CLCE parser to fail. The CLCE parser generated precise CGs, but it could not improve the output generated by Intellitex. A survey of these two systems can provide some insight into the issues.

The largest and most impressive application combined Intellitex with VAE for a legacy reengineering project that analyzed and related the software and English documentation for a large corporation. The software was written in formal languages: 1.5 million lines of COBOL with embedded SQL statements and several hundred scripts in the IBM Job Control Language (JCL). The documentation consisted of 100 megabytes of English reports, manuals, e-mails, web pages, memos, notes, and comments in the COBOL and JCL code. Some of the documentation and programs were up to 40 years old and still in daily use.

The first goal was to analyze the programs to derive a data dictionary, data flow diagrams, process architecture diagrams, and system context diagrams for all the software. That task could be done with programming-language parsers and conventional methods of analysis. The second and more challenging goal was to analyze the English, detect any errors or discrepancies between the software and the documentation, and generate a humanly readable glossary of terminology for the software and data, including all the variations over the period of 40 years. A major consulting firm estimated that analyzing all the documentation and relating it to the software would require 40 people for 2 years.

By using Intellitex and VAE, two programmers, Arun Majumdar and André LeClerc, accomplished both goals in less than two months, a total of 15 person weeks instead of 80 person years (LeClerc & Majumdar 2002). The results of first analyzing the computer languages and translating them to conceptual graphs were essential for analyzing the English. The names of every program, file, and data element were added to the dictionary used for parsing English. Furthermore, those items were also classified in an ontology of the computer terms that supplemented the ontology derived from WordNet, CoreLex, and other resources. Each term added to the lexicon was associated with one or more conceptual graphs that showed the expected relations: for example, variables occur in programs, programs process files, and files contain data. When parsing English, Intellitex translated every phrase or sentence to a conceptual graph. Those CGs that did not refer to anything in the software were discarded, and the others were used to update a knowledge base of information about the software. The results of the analysis were presented in one CD-ROM: software diagrams, data dictionary, English glossary, and a list of inconsistencies between the software and the documentation.

The reason why Intellitex and VAE succeeded where many natural language processors failed is that it did not attempt to translate informal language to formal logic. Instead, it used the formal CGs derived from COBOL, SQL, and JCL as the background knowledge for interpreting English text and resolving ambiguities. In short, the results were generated by joining formal CGs according to formal rules in order to create a pattern that had a close match to the approximate CGs derived from the English sentences. As an example, the following paragraph is taken from the English documentation:

> The input file that is used to create this piece of the Billing Interface for the General Ledger is an extract from the 61 byte file that is created by the COBOL program BILLCRUA in the Billing History production run. This file is used instead of the history file for time efficiency. This file contains the billing transaction codes (types of records) that are to be interfaced to General Ledger for the given month. For this process the following transaction codes are used: 32 — loss on unbilled, 72 — gain on uncollected, and 85 — loss on uncollected. Any of these records that are actually taxes are bypassed. Only client types 01 — Mar, 05 — Internal Non/Billable, 06 — Internal Billable, and 08 — BAS are selected. This is determined by a GETBDATA call to the client file. The unit that the gain or loss is assigned to is supplied at the time of its creation in EBT.

The common words in this paragraph were found in the dictionary derived from WordNet. Other words such as BILLCRUA, GETBDATA, and EBT were derived from the previous analysis of the software. Those words caused VAE to bring associated CGs from the background knowledge.

The sample paragraph also illustrates how Intellitex can process a wide range of syntactic constructions with a rather simple grammar. A phrase such as “32 — loss on unbilled” is not part of any published grammar of English. When Intellitex found that pattern, it translated it to a rudimentary conceptual graph of the following form:

```
[Number: 32]→(Next)→[Punctuation: "—"]→(Next)→[Loss]→(On)→[Unbilled]
```

This graph was stored as a tentative interpretation with a low weight of evidence. But Intellitex found two more graphs, which VAE matched to this graph with a high weight of evidence. Therefore, this syntactic pattern became, in effect, a newly learned grammar rule with a familiar semantic pattern. Although that pattern is not common in the full English language, it is important for the analysis of at least one document. The uninformative relations labeled `Next` were supplemented with background knowledge derived from previously analyzed CGs that formed the best match to those rudimentary graphs.

This discussion illustrates one of the most important lessons learned from Intellitex and VAE: A formal representation is easier to derive by joining conceptual graphs from background knowledge than by limiting the analysis to the details found in the input sentences. In fact, the background knowledge can often correct typos and other careless mistakes in the input text. That process illustrates Peirce's point that the listener may derive "a more developed sign" than the speaker intended. It is colloquially called "reading between the lines." This principle was applied in another application of Intellitex for scoring and correcting student answers to examination questions. Instead of trying to understand every detail of the students' often cryptic and ungrammatical prose, VAE would match the approximate CGs derived from the student answers to previously derived CGs that were known to be correct or incorrect. The results had a high correlation with the scores assigned by experienced teachers.

For applications that require a precise representation in logic, a syntax-directed parser was used to translate CLCE to logic. Following is an example of medical English, as written by a physician:

> Exclude patients with a history of Asthma, COPD3, Hypotension, Bradycardia (heart block > 1st degree or sinus bradycardia) or prescription of inhaled corticosteroids.

No system available today can accurately translate this kind of language to any version of logic. But a person with medical expertise and some training in writing controlled English can learn to translate this text to the following CLCE statements:

> Define "x is bradycardia" as "either x is sinus bradycardia or (x is a heart block and x has a degree greater than 1)".
>
> If a patient x has a history of asthma, or x has a history of COPD3, or x has a history of hypotension, or x has a history of bradycardia, or (x is prescribed a drug y, and y is inhaled, and y is a corticosteroid), then x is excluded.

Although these statements can be read as if they were English, CLCE is actually a formal language that has a direct mapping to first-order logic. For somebody who knows the subject matter, reading CLCE requires little or no training. Learning to write CLCE, however, requires training, especially for people who have never taken a course in logic.

To make CLCE more "user friendly," additional grammar rules were added to catch typical errors and to introduce more natural ways of expressing various logical combinations. But as we continued to add rules and inferences, we ran into maintenance problems and interactions between the inferences that resulted in confusing, but consistent readings. Since we were already implementing a new parser to replace Intellitex, we decided to design the new parser to handle CLCE as one kind of language game that could be played with English. The new parser would translate CLCE sentences directly to logic. But instead of rejecting sentences outside the CLCE subset, it would use the methods designed to handle unrestricted English. Then it would translate any CG that was generated back to CLCE as an echo and ask for a confirmation of its accuracy. In effect, CLCE was no longer defined as the language accepted by the parser, but as the language generated as an echo.

7. Designing Robust and Flexible Systems

A computer system that truly understands language would have to address all the issues discussed in this article, perhaps with others that are still unknown. Following is a brief summary:

1. Language learning, by the individual or the species, is grounded in social interactions, and full language understanding must be integrated with social behavior and all the supporting mechanisms of perception, action, knowledge, and reasoning.
2. Wittgenstein was correct in rejecting his early view, as influenced by his mentors Frege and Russell, that logic is the foundation for language. As he said in his notebooks (*Zettel*), language

is "an extension of primitive behavior. (For our language game is behavior.)"

3. Instead of being the foundation for language, logic is one among many important games that can be played with the words and syntax of a natural language. Formal logic is an abstraction from those language games, not a prerequisite for them.
4. The elegant syntax of well-edited prose is another important language game, which is used in large libraries of valuable knowledge. But focusing on that game as the prototype of "language competence" is as misguided as privileging any other language game, such as poetry, prayer, casual gossip, technical jargon, or text messaging.
5. Syntactic parsers can be useful for many practical applications, but a rigid linkage of syntactic rules to semantic rules is too inflexible and fragile to support natural languages. The appropriate semantics cannot be determined without knowledge of the context and subject matter.
6. The information needed to understand a sentence can rarely be derived from just its words and syntax. Even when the syntax is unambiguous, background knowledge of the context and subject matter must be added to determine the referents, the exact word senses, and the speaker's intentions.

The new VivoMind Language Processor (VLP) is a modular, open-ended system designed to accommodate the features discussed in this article and others that remain to be invented. The processing is handled by a society of agents, which can dynamically reconfigure their interactions by the market-driven learning methods described in Section 5. New features can be handled by adding new agents to the society, and a failure of one or more agents causes a fail-soft degradation in capability, rather than a hard crash. The syntactic component of VLP generates conceptual graphs as dependency structures by techniques similar to a link-grammar parser (Sleator & Temperley 1993), but with an approach that is similar to the parallel and concurrent ParseTalk (Hahn et al. 1994). Instead of the object-oriented methods of ParseTalk, in which the calling program determines how an object is supposed to respond, the VLP agents have more freedom to make their own decisions. Many of the decisions use a consensus-based approach that combines the results of several agents.

The first major application of VLP was to analyze 79 documents in the geosciences domain. The articles, which ranged in size from 1 to 50 pages, described various sites of interest for oil and gas exploration. The documents were not tagged or annotated in any way, except for the usual formatting tags intended for human readability. The VLP system translated the texts to conceptual graphs, used the new Cognitive Memory system to index and store the graphs, and searched for analogies among the graphs that described various sites. When two sites were found to be similar, the system would state in English which aspects of one site corresponded to which aspects of the other site. A domain expert who examined the output found these side-by-side comparisons to be especially informative and useful.

Several different resources were used to provide lexical knowledge, but no attempt was made to merge all the information in a single lexicon. Such a merger was rejected for several reasons. First, many resources, such as WordNet, CoreLex, and Roget's Thesaurus, are so different in kind, in level of coverage, and in organization that the merger would be difficult to do and difficult to undo, if one resource or the other were inappropriate for a particular application or even a particular sentence. Second, resources are revised and updated at different times, and frequent updates to a large, merged resource would be inconvenient and error prone. Third, there is no need to merge the resources in advance, since the society of agents makes it easy to assign one agent to each resource, and that agent can contribute any relevant information from its resource whenever it seems useful. Finally, nobody knows what resources may become available for special domains and purposes, and keeping different ontologies in separate modules maximizes the options and flexibility. A voting mechanism among the agents enables them to accept or reject any contribution, depending on the current task and context.

To illustrate the operations of the agents, the following sentence was taken from one of the geoscience articles (Sullivan et al. 2005):

> The Diana field is situated in the western Gulf of Mexico 260 km (160 mi) south of Galveston in approximately 1430 m (4700 ft) of water.

In the first stage, agents for the lexical resources contribute information about the part of speech of each word, its associated concept type, and various formats for measures and other typical qualifiers. Any conflicts among the agents are resolved by voting. The result is

> entity(1, "Diana field") prep("in the") loc(1, western) loc(2, "Gulf of Mexico") measure(1, "260 kilometers") measure(2, "160 miles") loc(3,south) prep("of") loc(4,"Galveston") prep("in") qualifier(approximately) measure(3,"1430 meters") measure(4,"4700 feet") prep("of") entity(2, water).

Lexical information, context, heuristics, and domain knowledge contribute to the interpretation. Low-level morphology expands “ft” to “feet”. Unknown words or word groups, such as “Diana field” are assumed to be geophysical entities because they are not in the basic lexicons. Agents with knowledge of the geoscience domain would determine that Diana field is a reservoir and add the information entity([1],reservoir). In geoscience, a reservoir is made up of special kinds of rocks that trap natural gases or oils. More detailed domain knowledge, used for interpreting the semantics and pragmatics of a text, can be stated in *scenarios*, which represent aspects of Wittgensteinian language games. Those scenarios, which domain experts can write in CLCE, can explain how a reservoir is created geologically or how it is evaluated for business purposes — two different, but related language games played by different employees of the same company.

Detecting the purpose of parenthesized expressions is usually difficult or impossible by syntactic methods, and the semantic conventions tend to be highly idiosyncratic and ad hoc. The initial analysis, as shown above, lists measure(1, "260 kilometers") and measure(2, "160 miles") in the sequence in which they occurred. One of the agents detected that the two measures were approximately equal, but stated in different units. Therefore, it made an abductive inference that the parenthesized expression was intended to express equality, and it added the information equals([1,2], measure). To determine the attachment of prepositional phrases and other modifiers, VLP creates an intermediate structure with multiple links as tentative hypotheses. For this sentence, the syntax is sufficient to determine that Diana field must be in the western part of the Gulf of Mexico, but other attachments are syntactically ambiguous. To prune away unlikely options, some agents use a domain ontology for geoscience, which includes information about reservoirs, bodies of water, and cities. After pruning, the remaining links indicate that Diana field is south of Galveston and in the water.

The protocol for interactions among VLP agents is a subset of the FMF, implemented in an extended version of Prolog called PrologIKS. Since all messages are expressed in the same Prolog-like notation, one of the FMF features that is not required by VLP is the option of supporting multiple languages. Most of the inferences are performed directly with Prolog rules. Domain-dependent rules, written in CLCE by domain experts, can be translated to Prolog for use by VLP agents. The agents that apply domain knowledge use exactly the same protocols as the agents that implement the morphology, the grammar, the lexical knowledge, and the multiple ontologies. When conflicts arise, the voting mechanisms can accept inputs from all those agents to resolve ambiguities and choose alternative interpretations. The MDL learning mechanisms can shift the influence of various agents by shifting the amount of resources allocated to them. If an agent’s contribution leads to a successful parse and interpretation of a sentence, it is rewarded with more resources. When analyzing different genres, documents, or even paragraphs, MDL methods can dynamically adjust the influence of different agents.

The VLP parser is still in an early stage of development, but it has produced some useful results. The market-driven learning methods have proved to be successful on another project, but they haven't yet been extensively tested on VLP. The semantic distance measures of the old VAE were highly efficient for analogy finding, and variations were applied to knowledge capture (Majumdar et al. 2007). More research and testing is necessary, but the new VLP already appears to be more robust, scalable, and accurate than the old Intellitex.

References

Deacon, Terrence W. (1997) *The Symbolic Species: The Co-evolution of Language and the Brain*, W. W. Norton, New York.

Delavenay, Emile (1960) *An introduction to machine translation*, Thames & Hudson, London.

Eberbach, Eugene, Dina Goldin, & Peter Wegner (2004) "Turing's Ideas and Models of Computation," in C. Teuscher, ed., *Alan Turing: Life and Legacy of a Great Thinker*, Springer, Berlin.

Frege, Gottlob (1879) *Begriffsschrift*, English translation in J. van Heijenoort, ed. (1967) *From Frege to Gödel*, Harvard University Press, Cambridge, MA, pp. 1-82.

Gelernter, David (1985) "Generative communication in Linda," *ACM Transactions on Programming Languages and Systems*, pp. 80-112.

Good, Irving John (1965) "Speculations concerning the first ultraintelligent machine" in F. L. Alt & M. Rubinoff, eds., *Advances in Computers* **6**, Academic Press, New York, 31-88.

Greenspan, Stanley I., & Stuart G. Shanker (2004) *The First Idea: How Symbols, Language, and Intelligence Evolved from Our Primate Ancestors to Modern Humans*, Da Capo Press, Cambridge, MA.

Hahn, Udo, Susanne Schacht, & Norbert Broker (1994) "Concurrent Natural Language Parsing: The ParseTalk Model," *International Journal of Human-Computer Studies* **41**, 179-222.

Hattiangadi, Jagdish N. (1987) *How is Language Possible? Philosophical Reflections on the Evolution of Language and Knowledge*, Open Court, La Salle, IL.

ISO/IEC (2007) *Common Logic (CL) — A Framework for a family of Logic-Based Languages*, IS 24707, International Organisation for Standardisation.

Kamp, Hans (2001) "Levels of linguistic meaning and the logic of natural language," http://www.illc.uva.nl/lia/farewell_kamp.html

Kittredge, Richard, & John Lehrberger, eds. (1982) *Sublanguage: Studies of Language in Restricted Semantic Domains*, de Gruyter, New York.

LeClerc, André, & Arun Majumdar (2002) "Legacy revaluation and the making of LegacyWorks," *Enterprise Architecture* **5:9**, Cutter Consortium, Arlington, MA.

Limber, John (1973) "The genesis of complex sentences," in T. Moore, ed., *Cognitive Development and the Acquisition of Language*, Academic Press, New York, 169-186.

Majumdar, Arun, Mary Keeler, John Sowa, & Paul Tarau (2007) "Semantic distances as knowledge capture constraints," *Proc. First International Workshop on Knowledge Capture and Constraint Programming*.

Milner, Robin (1999) *Communicating and Mobile Systems: the π calculus*, Cambridge University Press, Cambridge.

Minsky, Marvin (1987) *The Society of Mind*, Simon & Schuster, New York.

Minsky, Marvin (2006) *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*, Simon & Schuster, New York.

Mithen, Steven (2006) *The Singing Neanderthals: The Origin of Music, Language, Mind, and Body*, Harvard University Press, Cambridge, MA.

Mohanty, J. N. (1982) *Husserl and Frege*, Indiana University Press, Bloomington.

Ogden, C. K., & I. A. Richards (1923) *The Meaning of Meaning*, Harcourt, Brace, and World, New York, 8th edition 1946.

Peirce, Charles S. (1902) *Logic, Considered as Semeiotic*, MS L75, edited by Joseph Ransdell, http://members.door.net/arisbe/menu/library/bycsp/l75/l75.htm

Peirce, Charles Sanders (1911) “Assurance through reasoning,” MS 670.

Peirce, Charles Sanders (CP) *Collected Papers of C. S. Peirce*, ed. by C. Hartshorne, P. Weiss, & A. Burks, 8 vols., Harvard University Press, Cambridge, MA, 1931-1958.

Selfridge, Oliver G. (1959) “Pandemonium: A paradigm for learning,” in *The Mechanization of Thought Processes*, NPL Symposium No. 10, Her Majesty’s Stationery Office, London, pp. 511-526.

Sleator, Daniel, & Davy Temperley (1993) “Parsing English with a Link Grammar,” *Third International Workshop on Parsing Technologies*, http://www.cs.cmu.edu/afs/cs.cmu.edu/project/link/pub/www/papers/ps/LG-IWPT93.pdf

Sowa, John F. (2002) “Architectures for intelligent systems,” *IBM Systems Journal* **41:3**, 331-349.

Sowa, John F. (2004) “Graphics and languages for the Flexible Modular Framework,” in K. E. Wolff, H. D. Pfeiffer, & H. S. Delugach (2004) *Conceptual Structures at Work*, Proceedings of ICCS 2004, LNAI 3127, Springer, Berlin, pp. 31-51.

Sowa, John F., & Arun K. Majumdar (2003) “Analogical reasoning,” in A. de Moor, W. Lex, & B. Ganter, eds. (2003) *Conceptual Structures for Knowledge Creation and Communication*, LNAI 2746, Springer, Berlin, pp. 16-36.

Sullivan, Morgan D., J. Lincoln Foreman, David C. Jennette, David Stern, Gerrick N. Jensen, Frank J. Goulding (2005) “An Integrated Approach to Characterization and Modeling of Deep-water Reservoirs, Diana Field, Western Gulf of Mexico,” *Search and Discovery*, Article #40153.

Thouin, B. (1982) “The METEO system,” in V. Lawson, ed., *Practical Experience of Machine Translation*, North-Holland, Amsterdam, pp. 39-44.

Wang, Hao (1960) “Toward mechanical mathematics,” *IBM Journal of Research and Development* **4**, pp. 2-22. http://www.research.ibm.com/journal/rd/041/ibmrd0401B.pdf

Warren, R. M. (1970) “Restoration of missing speech sounds,” *Science* **167**.

Wildgen, Wolfgang (1982) *Catastrophe Theoretic Semantics: An Elaboration and Application of René Thom’s Theory*, John Benjamins Publishing Co., Amsterdam.

Wildgen, Wolfgang (1994) *Process, Image, and Meaning: A Realistic Model of the Meaning of Sentences and Narrative Texts*, John Benjamins Publishing Co., Amsterdam.

Wittgenstein, Ludwig (1921) *Tractatus Logico-Philosophicus*, Routledge & Kegan Paul, London.

Wittgenstein, Ludwig (1953) *Philosophical Investigations*, Basil Blackwell, Oxford.

Wittgenstein, Ludwig (1970) *Zettel*, University of California Press, Berkeley.