METAPHOR AND SYMBOLIC ACTIVITY *11*(1), 85–100

Computational Approaches to Figurative Language

James H. Martin
University of Colorado

Computational approaches to figurative language are concerned with those situations in which the appropriate conceptual representation of an utterance differs fundamentally from that which could be composed from the ordinary meanings of the utterance's constituent words. The bulk of computational work in this area has addressed idiom, metonymy, metaphor, and indirect speech acts. These phenomena are generally recognized as being major impediments to the construction of successful natural language processing systems. Most such systems have attempted to deal with such figurative language through the use of extended notions of language conventions, general purpose reasoning methods, and strong notions of contextual expectations.

Computational approaches to figurative language are concerned with those situations in which the appropriate conceptual representation of an utterance differs fundamentally from that which could be composed from the ordinary meanings of the utterance's constituent words. Although there are a large number of figurative language devices that lead to such situations, the bulk of the computational work in this area has addressed idiom, metonymy, metaphor, and indirect speech acts. These phenomena are generally recognized as major impediments to the construction of successful natural language processing systems.

Such language is, of course, ubiquitous and, for the most part, easily handled by people. The question for those engaged in computational modeling is, if the standard resources and techniques for composing representations are inadequate for figurative language, then what other resources and techniques should be brought to bear? Computational researchers have, for the most part, drawn inspiration for their models from various schools of research in the cognitive sciences. Broadly speaking, these approaches have

Requests for reprints should be sent to James H. Martin, Computer Science Department and Institute of Cognitive Science, University of Colorado, Boulder, CO 80309–0430.

fallen into three categories: convention-based approaches, reasoning-based approaches, and context-based approaches.

Convention-based approaches rely on some explicitly enumerated set of facts that allow a language processor to associate one type of representation with another. Such approaches can be seen as extensions of the traditional Saussurean form-meaning model. In the case of idiom, the extension allows the association of multiword forms with specific meanings possibly unrelated to the meanings of the individual words. In the case of metaphor, metonymy, and speech acts, the extension has been more radical: the direct structured association of specific concepts with other concepts.

Reasoning-based approaches eschew the notion of figurative language conventions altogether. Rather, these approaches focus on what can be computed from the conceptual representation formed compositionally from the utterance. That is, there is assumed to be some procedure by which a strictly compositional meaning is taken as input and a new more appropriate representation is produced as output. Work in this vein often takes the view that figurative language processing is best approached as a problem for a general cognitive processing faculty rather than as a specifically language-related phenomenon. It follows, therefore, that the computational models following this approach are often based on preexisting models intended for more general reasoning tasks.

The bulk of the work in this vein has been on metaphor and indirect speech acts. In the case of metaphor, the reasoning is typically accomplished by some form of analogical reasoning system. Speech act processing is normally accomplished through the use of a general purpose rule-based reasoning system that uses the participants' shared knowledge of their mutual plans and goals.

Finally, by context-based approaches, I mean those approaches in which context has the primary role of proposing plausible meanings rather than merely selecting among meanings generated by other methods. Such approaches assume that figurative uses occur within a richly structured current context. This context typically includes a set of concepts created, or activated, by the previous discourse. This extended notion of context is usually assumed to include a likely meaning for a subsequent figurative utterance. In these approaches, the actual sentence "meaning" of an utterance is best seen as a cue to be used to select and possibly flesh out some part of a current discourse representation. The inspiration for this work tends to come from two areas of research in cognitive psychology: studies concerning the time needed to comprehend figurative language in context and research on discourse processing that focuses on how people use inference to construct conceptual representations of text.

The remaining sections of this article present overviews of various computational approaches to idiom, metonymy, metaphor, and indirect speech acts organized along the lines of these three general approaches. Before describing these efforts, I will briefly sketch in the next section what is usually assumed to be involved in the garden-variety semantic interpretation of literal language in natural language processing research. I should note that neither this description nor any of the descriptions of actual figurative language systems will be at a particularly technical level. Readers interested in the details of computational semantic analysis should consult Allen (1995), Hirst (1987), and Gazdar and Mellish (1989). Recent collections addressing the current technical aspects of figurative language processing can be found in Fass, Hinkelman, and Martin (1991, 1992), Barnden and Holyoak (1994), and Pustejovsky and Bergler (1992).

A BRIEF SKETCH OF COMPUTATIONAL SEMANTICS

In the vast majority of computational semantic research, the lexicon is taken to be a simple repository of mappings from forms to some kind of underlying meaning representations. In most models, the representation underlying nouns amounts to some type of attribute value list, often augmented with a type hierarchy. Verbs are typically treated as predicates with a fixed set of typed argument slots following some version of a deep case representation (Fillmore, 1968). The conceptual analysis of a single utterance consists of composing a representation from the representations associated with the individual constituent words according to composition rules either explicitly or implicitly associated with the syntactic structure of the sentence.

The principal problem that has to be addressed by such systems is controlling the considerable ambiguity that is inherent in the approach. This ambiguity arises from the large number of homonymous and polysemous word senses as well as from the grammar itself. Without some method of controlling this ambiguity, a natural language processing system would be faced with hundreds of representations for a single utterance. The most popular methods of dealing with ambiguity use a combination of semantic selection restrictions on the arguments to predicates, real-world commonsense knowledge, and knowledge of the immediate discourse context. It is hoped that, when combined, these techniques will be sufficient to select a single appropriate representation from the range of possible ones that might be created compositionally.

Unfortunately, in the case of figurative language, this approach usually winds up eliminating all candidate meanings because the figurative nature of the input leads them to be ill-formed in most cases. Of course, because the filtering of ill-formed representations is what these systems rely on to control ambiguity, the pervasive use of figurative language poses a major theoretical and practical hurdle to the ultimate success of these systems.

IDIOM

Idioms are the simplest and earliest form of figurative language to be addressed by the computational community. Unfortunately, they have also proven to be difficult for a number of reasons, but mainly because most idioms do not adhere to the simple notion of a fixed sequence of words associated with a fixed meaning. Instead, they display a fascinating array of syntactic and semantic variety that makes it difficult to come up with a unified computational framework capable of dealing with all types of idioms.

The bulk of computational work specifically addressing idiom has been driven by the need to account for this observed productivity (Becker, 1975; Chin, 1992; Jurafsky, 1993; Lytinen, Barridge, & Kirtner, 1992; Stock, 1989; van der Linden, 1992; Wilensky & Arens, 1980; Zernik, 1987; Zernik & Dyer, 1987). To illustrate some of the issues involved, consider the following examples:

"John and Mary tripped the light fantastic."
"John and Mary buried the hatchet."
The more I drink the sillier I get.

The *"tripped the light fantastic"* use is a clear example of the simplest form of idiom: Its meaning seems to have no relation to any of the constituent parts, it resists any kind of syntactic or semantic productivity, and it is distinctly at odds with the normal constraints of English grammar. Idioms such as this correspond most closely to the simple notion of an essentially fixed and arbitrary sequence of words with a special meaning. Computationally, such idioms are dealt with by simply associating this meaning with this fixed sequence of parts and indexing the pattern under an appropriate grammatical category. In addition, in cases like this, there is no ambiguity problem because there is no valid literal interpretation.

Unfortunately, one does not have to look far to find example idioms that violate one or all of these typical characteristics. Consider the *"bury the hatchet"* example. The source of difficulty posed by this idiom is that there seems to be an intuition that the notions of "burying" and "hatchets" play some deep, if unclear, role in the meaning of this construction (Gibbs, 1989). This intuition becomes important when one considers the syntactic productivity of the construction. For the *"bury the hatchet"* idiom, this degree of compositionality seems to be enough to sanction the well-formedness of the passive version of the idiom. This is as opposed to the otherwise similar *"kick the bucket,"* where the bucket seems to play no underlying role and correspondingly exhibits no productivity.

Despite a somewhat bewildering array of different formalisms, nearly every currently implemented computational system that addresses this issue

(Chin, 1992; Jurafsky, 1993; Lytinen et al., 1992; Wilensky & Arens, 1980; Zernik & Dyer, 1987) makes the same claim. The syntactic productivity of a stored idiom arises from the association of parts of the surface pattern with parts of the underlying conceptual representation of the idiom's meaning. In some cases, as in Zernik and Dyer's RINA system, a further attempt is made to motivate the association between the part of the pattern and the underlying role. In this case, something akin to the notion that hatchets are weapons and burying them prevents their use would constitute such an explicit motivation.

Finally consider the *"the more I drink"* example. In their *construction grammar* approach, Fillmore, Kay, and O'Connor (1988) dubbed the underlying motivation for this use the "the Xer the Yer" construction. The only lexically specified parts of this idiomatic construction are the two uses of *the.* The remaining components are defined via a complex mixture of syntactic, semantic, and pragmatic parts. Current computational approaches to construction grammar (Jurafsky, 1993) are concerned with coming up with knowledge representation formalisms expressive enough to capture this complex mixing of constraints that, at the same time, still allow the implementation of efficient conceptual analysis programs.

In addition to these attempts to account for idiomatic generativity in a more principled way, recent work (Jurafsky, 1993; Lytinen et al., 1992; van der Linden, 1992) also began to focus on the psychological plausibility of the underlying representations and their associated processing models. This work is driven by the wealth of empirical timing data now available about human processing of idiomatic language (Cacciari & Tabossi, 1988; Gibbs & Nayak, 1989). These models attempt to accurately predict the initial time of activation of an idiomatic construction as well as the point at which the evidence is sufficient to select the idiom over competing analyses.

METONYMY

Metonymy has long been recognized as a problem for natural language processing systems. The obvious approach, not unlike similar approaches to idiom, is to simply list all possible senses of those words that are frequently used metonymically. For example, a word such as *newspaper* would simply have a large number of distinct senses to account for the following uses.

Today's newspaper is in my driveway.
Did you read today's newspaper?
Don't believe what you read in the newspaper.
Mary's father works for the newspaper.
The local newspaper is located downtown.

A lexicalist approach would simply list distinct senses for the physical copy of a single newspaper, an edition, a kind of medium, an organization that publishes a newspaper, and its location. This is somewhat problematic because it leads to an impractically large lexicon that, at the same time, is inadequate because there is no mechanism for dealing with unanticipated metonymic uses (i.e., a lexicon that is large and often does not work).

In addition to its practical shortcomings, this approach fails to capture any of the systematicity underlying most uses of metonymy. For example, the word *magazine* displays many, if not all, of the aforementioned senses of *newspaper*. Coming up with a way to exploit this type of systematicity is the motivation behind most computational approaches that directly address metonymy.

The META5 system (Fass, 1988, 1991) attempts to capture these systematicities through the use of a set of rules. These rules allow the system to specify that certain conceptual categories can be used to refer to other related categories. For example, to handle well known *"AUTHOR FOR WORK"* examples such as *"Plato is on the top shelf,* Fass's system simply encodes that schema as a rule. During interpretation, the mention of an author can then be used to produce a representation of the author's work. This new representation is placed into whatever conceptual role the author played in the utterance. To control its application, this mechanism is only triggered by the presence of some kind of semantic violation in the original utterance. Rules applicable to the concept causing the violation are then fired in an attempt to find another concept that alleviates the violation.

Pustejovsky's (1991) work on the *generative lexicon* represents another attempt to account for the systematicity underlying many metonymic uses. This work attempts to make the lexicon a more active generative device rather than a passive list of words. More specifically, the semantic representation of a noun in a given utterance is viewed as having been generated during the interpretation process rather than merely looked up. The generated representation is produced by combining information about the conceptual type of the noun with the constraints being imposed on it by the head of its verb phrase. In such an approach, metonymy is not an anomalous situation to be dealt with by a special mechanism. Rather, the interpretation of all nouns requires this type of sentence-specific sense generation. This approach is somewhat similar to Fass's (1988, 1991) approach in that it does, in effect, enumerate the functional roles that nouns may take on in combination with verbs.

Unfortunately, neither of these approaches are sufficient to deal with nonconventional metonymic uses such as the following.

"The ham sandwich didn't leave a tip."

It would be implausible for Fass's (1988, 1991) META5 system to have a preexisting *"ORDERED DISH STANDS FOR CUSTOMER"* rule in its knowledge base. Similarly, Pustejovsky's (1991) theory explicitly excludes this type of contextually motivated metonymy from the purview of the generative lexicon, leaving it to some more general discourse reasoning capability.

The Metallel system (Iverson & Helmreich, 1992), on the other hand, attempts to provide a mechanism that can deal directly with this type of metonymy. In fact, Iverson and Helmreich went further and argued that metonymic conventions of any type are unnecessary, because if some other more general reasoning mechanism is needed to deal with open-ended context-generated metonymys, that same general mechanism can deal with the more mundane examples as well. In the Metallel system, this is accomplished via a limited spreading activation technique. Spreading activation refers to a method whereby a system accesses concepts directly and indirectly related to those mentioned directly in a sentence.

Metallel uses spreading activation to implement an intersection search through its knowledge base of facts in an attempt to determine an appropriate referent for some noun phrase. In the ham sandwich case, the system would presumably find that the best connection in context from ham sandwich to tipping was through the customer concept, concluding that the customer who had ordered the ham sandwich was also doing the tipping.

As we will see with similar approaches to metaphor, this approach is prone to the overgeneration problem. Systems like Metallel exploit any of the available domain relations between the source and target concepts of the metonymy. However, the system falls short as an explanatory theory of metonymy because only a small fraction of such relations ever give rise to well-formed metonymies.

METAPHOR

Although metaphor has long been of concern to researchers in natural language processing (DeJong & Waltz, 1983; Russell, 1976; Wilks, 1975, 1978), much of current interest in the topic can be traced to the influence of Lakoff and Johnson's (1980) *Metaphors We Live By.*

Convention-Based Approaches to Metaphor

Taking direct, and sometimes overly literal, inspiration from this work, there are by now a large number of convention-based approaches to metaphor interpretation and generation (Hayes & Bayer, 1991; Jacobs, 1987; Jones & McCoy, 1992; J. Martin, 1990, 1992; Norvig, 1987, 1989; Veale & Keane, 1992). These approaches make the assumption that some core set of conceptual metaphors should be explicitly represented as part of a system's knowledge. To illustrate how such conventions can be exploited computationally, consider the following example.

"IBM got out of the typewriter business."

This example is based on a common conceptual metaphor that allows activities to be viewed as containers. In this instance, it makes use of the notion entailed by this metaphor that exiting an activity viewed as a container implies ceasing participation in the activity. More computationally, such a rule would explicitly link facts associated with physical containers with otherwise separate facts about activities.

When a concept is created compositionally that matches the source part of such a conventional metaphorical rule, the rule is used to create a new concept as specified by the target part of the rule. For this example, the semantic interpreter would produce a representation specifying something akin to a physical exiting of an physical object from a container. A concept representing IBM would fill the role of the physical object, whereas a representation of "the typewriter business" would play the role of the container. This skeletal representation would, in turn, activate an appropriate set of metaphorical association rules, which would, in turn, be used to produce a representation that replaces the notion of exiting with one capturing the notion of ceasing participation in some activity.

Of course, the various computational mechanisms embodying this approach instantiate these notions in a number of different ways. The primary differences involve the amount of structure that needs to be explicitly captured in the representations and the specificity of the encoded metaphors. These issues reveal the differing ways that computational researchers have interpreted the claims made by Lakoff and Johnson (1980).

One such view is that our conceptual system is structured by an extremely small number of abstract, loosely structured schemas. This view typically focuses on abstract orientational and body-centered metaphors. Veale and Keane (1992) coined the term *conceptual scaffolding* to refer to this view. Under this approach, a system would be expected to need an extremely small number of these abstract metaphorical schemas. During the process of interpretation, these schemas only provide the scaffolding for the final representation. The more detailed representations required for actual domain applications would be produced by augmenting these initial representations with long-term facts about the target domain and context specific inferences. Such an approach has the immediate practical advantage of requiring very little in the way of knowledge engineering. Natural language processing applications need only be provided with a small set of presumably universal metaphorical schemas to handle a wide range of metaphorical language.

At the other end of the spectrum is the work of J. Martin (1994) and Jones and McCoy (1992). Both of these efforts assume the need for a much larger number of highly structured metaphor schemas organized into some kind of hierarchical structure. Much of this work is driven by the desire to have the metaphorical knowledge base bear the primary burden of accounting for the wide range of metaphors observed in real texts.

In J. Martin's (1994) METABANK work, this desire arises directly from empirical research performing a metaphorical analysis of a large corpus. J. Martin claimed that an extremely small knowledge base of highly abstract orientational metaphors is simply not sufficient to account for the range of metaphors that are actually encountered in practice. In most genres of text, there appear to be a large number of apparently idiosyncratic domain-to-domain schemas. Although such metaphors have clear motivations based on more abstract metaphorical schemas, their existence in a given domain is just as clearly not predictable from the abstract schemas alone.

The work by Jones and McCoy (1992) followed a similar path from a more practical point of view. Jones and McCoy directly addressed the issue of generating metaphors from underlying semantic representations. Their concern was driven by the generative power of more abstract mappings. Simply put, they predicted a large number of metaphors that are never seen and would likely be judged anomalous by hearers. To ensure that their Quipper system only generates well-formed metaphors, Jones and McCoy made use of metaphors that are fairly specific and highly structured.

Reasoning-Based Approaches to Metaphor

Reasoning-based approaches to metaphor attempt to exploit information about the source and target concepts of a metaphor to determine an appropriate meaning without using prior knowledge of metaphorical conventions. The dominant paradigm in this area is the analogical approach (Carbonell, 1982; Fass, 1988, 1991; Gentner, 1983; Gentner, Falkenhainer, & Skorstad, 1988; Indurkhya, 1987; Weber, 1994). This approach asserts that metaphor depends on inherent structural similarities between the disparate source and target domains. It is the job of an analogical reasoning engine to uncover and specify the exact nature of the correspondences.

The simplest approaches address the comprehension of metaphors that take the form of *"X is a Y."* The source and target concepts are fully specified to the model. Following traditional views of metaphor, the task of the model is to use its knowledge of the source and target domains to alter the representation of the target concept. This process has typically involved one of three operations: changing the value of an existing feature in the target, changing the level of salience or prominence of a feature of the target, or transferring a feature–attribute pair from the source to the target (Chandler, 1991; Weiner, 1984).

Although theoretically interesting, this type of metaphor has an extremely low frequency in actual text. More critical to natural language processing applications are analogical approaches that directly address the more frequent relational, or predicate, kind of metaphor. Consider the previous container metaphor again:

"IBM got out of the typewriter business."

The first heuristic used in most systems is that the predicate-like term is interpreted as belonging to the source domain with some or all of its arguments belonging to the target domain. This heuristic seems to hold for a large number of examples and has some psycholinguistic support (Gentner et al., 1988) but is by no means inviolable, as can be seen in examples like *"John married a gem."*

The task of an analogical metaphor interpreter is to find or create the correct set of relations linking IBM to the typewriter business by making reference to concepts relating to getting out of containers. This is done without making use of any explicit associations between these domains.

This involves establishing a correspondence between the two domains. A significant part of this step is given by the predicate-argument structure of the original input. In our example, syntactic knowledge combined with case-role information about "get out of" tells us that IBM corresponds to the expected enterer, with the typewriter business playing the thing gotten into. Schematically, this information leaves us with the following analogical formula:

physob—exit—container::IBM—???—typewriter-business

The principle remaining task for the analogy engine is to find the missing concepts that will link up the specified pieces of the target concept in a way that most harmoniously satisfies the constraints imposed by this metaphorical equation. Of course, in any reasonably robust knowledge base there will be a large number of concepts that could relate the concepts of IBM and the typewriter business. This situation often leads to an exponentially large number of candidate analogical matchings between the source and target domains. The principal difference among the various analogical approaches to metaphor is the method they use to assess the well-formedness of candidate matchings.

The work by Gentner and colleagues (Gentner, 1983; Gentner et al., 1988) used her *systematicity principle* to judge the well-formedness of candidate solutions. This principle favors those solutions that preserve interrelated sets of relations among the various source and target concepts. In this scheme, this formalized notion of interrelatedness takes precedence over solutions based on features of individual objects or solutions based on matching individual relations. Indurkhya (1987) offers an alternative structural approach with different well-formedness criteria.

Fass (1991), in his work on META5, focused on the need to match different relations from the source and target domains. Unlike much of the work on analogy, Fass did not assume that the names of relations used to structure disparate domains will ever match exactly. Rather, relations, like other concepts, must be arranged into a hierarchy. META5 is based on a hierarchical matching scheme in which matches between relations are based on where in the abstraction hierarchy a common ancestor relation is found. Candidate solutions are then evaluated based on the degree of abstractness and interrelatedness of component relations in the differing solutions.

Carbonell (1982) took an approach that focuses on the content rather than on the structure of analogical solutions. This approach is based on the notion that certain relations should be privileged with respect to analogical matching. Carbonell suggested that a prioritized list of favored relations should be consulted when evaluating possible matches.

The primary tension between all of these approaches and approaches based on metaphorical conventions arises from the nature of the representation of many target domains, particularly those that seem to have an impoverished semantics absent some conventional metaphor. Consider the well-studied *"LIFE IS A JOURNEY"* metaphor. In the analogical-reasoning view, there must be a preexisting similarity between the conceptual representations of *"JOURNEYS"* and *"LIFE."* This similarity forms the basis for the metaphor and is discovered during the interpretation process. We can, however, ask whether it is not the case that our conception of the category *"LIFE"* is itself structured via the *"JOURNEY"* concept. If this is the case, then analogical approaches are attempting to rediscover the structural similarities between domains when one was created from the other as a template. Approaches based on convention take it to be more sensible to directly represent that history in the conceptual category structure than to rediscover it with each instance.

The most recent work by Indurkhya (1992) addressed this issue from a developmental viewpoint. The role of the analogy engine is shifted from the point of view of a competent processor of metaphors to the role of a developmental mechanism. In this view, the critical task for the analogy engine lies in the metaphorical creation of target concepts from well-specified source concepts. In his view, both the convention-based approach and the analogical-matching approach beg the question of how the target concept gets structured the way it does. The bulk of Indurkhya's computational modeling work has focused on the problem of metaphors with some visual basis.

Context-Based Approaches to Metaphor

As with most other issues in conceptual analysis, the context within which a metaphor is produced has a profound effect on its meaning. A number of researchers have made context the starting point for their theories of metaphor processing (Hobbs, 1979; C. Martin, 1993). They are supported by a wealth of psycholinguistic data showing significant processing differences for metaphors encountered in supporting contexts as opposed to those presented in isolation. The vast majority of these efforts attempt to show that the processing of metaphor can be made to fall out from mechanisms already needed to deal with nonmetaphorical text.

Note that these efforts do not merely make the assumption, common to nearly all computational models, that context is needed to select among plausible meanings proposed by some other context-independent semantic processor. Rather, they assert that context has the primary role of suggesting plausible representations. Under this framework, the semantic content of a new sentence is viewed as a set of clues that are to be used to find an appropriate way to extend the existing representation of the context. In the case of Hobbs (1979), this was accomplished by a limited context-directed inferencing system. C. Martin's (1993) direct memory access parsing (DMAP) system is based on an example-based approach that links metaphors in new utterances to specific remembered contexts of past use.

INDIRECT SPEECH ACTS

Work on speech act processing arises from the importance of discerning speaker's goals while they are interacting with various natural language processing applications. Consider the following example from a help system for the UNIX operating system:

> *"Can you tell me how to edit a file?"*
> *"Do you know how to delete a file?"*

The literal, or compositional, interpretations of these utterances clearly do not capture what the user is really trying to convey. As with other figurative approaches, computational research into this phenomenon has been divided into two camps: The first camp asserts that an appropriate meaning can be derived by reasoning about the domain semantics; the second asserts that specialized knowledge about the nature of specific speech acts must be employed.

Reasoning-based approaches to indirect requests have focused on the use of knowledge about the shared plans and goals of question askers and answerers (Allen & Perrault, 1980; Cohen & Perrault, 1979). In particular, users of such systems are assumed to have the goal of finding out how to do things with the system. More specifically, it can be assumed that they intend to acquire this knowledge by asking questions, the answers to which will provide the needed information.

Such approaches have been problematic for two reasons: They are quite costly in terms of the amount of time needed to compute what should be an easily derived meaning, and they suffer from the overgeneration problem. Specifically, they fail to tightly predict the restricted lexical and grammatical forms that are typically used with such speech acts.

Paralleling work on metaphor and metonymy, convention-based approaches to indirect requests come in two varieties: conceptual associations and surface-level constructions (Hinkelman, 1989; Hinkelman & Allen, 1989; Mayfield, 1992). Conceptual associations for speech acts are similar to those employed for conventional metaphors. Representations of the literal meanings are explicitly linked to the appropriate meanings via direct associations. For example, the construction underlying the previous examples would state that asking about someone's prerequisite ability to perform an action is a way of asking them to explain the action. This single rule accounts for both examples.

Of course, such an approach immediately leads to an ambiguity problem because both the literal meaning and the figurative meaning are possible. Both Mayfield (1992) and Hinkelman and Allen (1989) addressed computational ways to deal with this situation in the context of interactive natural language help systems.

Unfortunately, these conceptual conventions suffer from the usual overgeneration problem. Consider the following examples:

"Can you tell me how to delete a file?"
"?Are you able to tell me how to delete a file?"

The second example corresponds to many of the schemas suggested for handling the first example. It is, however, clearly ill-formed in some way and simply does not occur with the indirect meaning. Hinkelman's (1989) thesis work addressed this problem by adding a mix of grammatical and semantic constraints to the convention, thereby constraining the range of productivity of its use.

SUMMARY

The earliest computational work on figurative language can be characterized as an attempt to simply get something to work within some existing application framework. Getting something to work meant coming up with some scheme in which some representation judged appropriate to an application could be produced from what would otherwise have to be considered an ill-formed input. These schemes, when viewed from either a theoretical or practical perspective, were found wanting for several reasons including their failure to hold up when the issue of productivity or generativity was considered, their failure to match up well with known psycholinguistic constraints concerning reaction times, and their failure to adequately account for the influence of context. Contemporary computational systems have had considerable success in addressing many of these issues. There remains, however, a distinct lack of consensus in the field as a whole. There has, in particular, been little movement toward integrating successful techniques from the various approaches into a unifying framework.

It is important to note that these computational efforts have derived considerable benefit from paying close attention to empirical psycholinguistic constraints concerning timing data. Further progress may be made toward a unifying framework if other empirical constraints are taken seriously. Three sources for such constraints are ripe to be exploited: corpus-based data concerning naturally occurring figurative uses in context, cross-linguistic data becoming available through the efforts of those working on machine translation, and, finally, the considerable developmental data now available as a part of the Childes project (MacWhinney, 1991).

REFERENCES

Allen, J. (1995). *Natural language understanding* (2nd ed.). Redwood City, CA: Benjamin-Cummings.

Allen, J., & Perrault, C. (1980). Analyzing intention in utterances. *Artificial Intelligence, 15,* 143–178.

Barnden, J., & Holyoak, K. (Eds.). (1994). *Analogy, metaphor and reminding.* Norwood, NJ: Ablex.

Becker, J. (1975). *The phrasal lexicon.* Paper presented at the Workshop on Theoretical Issues in Natural Language Processing, Cambridge, MA.

Cacciari, C., & Tabossi, P. (1988). The comprehension of idioms. *Journal of Language and Memory, 27,* 668–683.

Carbonell, J. (1982). Metaphor: An inescapable phenomenon in natural language comprehension. In M. Ringle & W. Lehnert (Eds.). *Strategies for natural language processing* (pp. 415–434). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Chandler, S. (1991). Metaphor comprehension: A connectionist approach to implications for the mental lexicon. *Metaphor and Symbolic Activity, 6,* 227–258.

Chin, D. (1992). Pau: Parsing and understanding with uniform syntactic, semantic and idiomatic representations. *Computational Intelligence, 8,* 456–476.

Cohen, P. R., & Perrault, C. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science, 3,* 177–212.

DeJong, G. F., & Waltz, D. L. (1983). Understanding novel language. *Computers and Mathematics with Applications, 9,* 131–147.

Fass, D. (1988). *Collative semantics: A semantics for natural language* (CRL Rep. No. MCCS–88–118). Lac Cruces: New Mexico State University.

Fass, D. (1991). met*: A method for discriminating metaphor and metonymy by computer. *Computational Linguistics 17,* 1, 49–90.

Fass, D., Hinkelman, E., & Martin, J. (Eds.). (1991). *Proceedings of the IJCAI Workshop on Compuational Approaches to Non-Literal Language: Metaphor, metonymy, idiom, speech acts and implicature.* (Tech. Rep. No. CU–CS–550–91). Boulder: University of Colorado, Computer Science Department.

Fass, D., Hinkelman, E., & Martin, J. (Eds). (1992). Non-literal language [Special issue]. *Computational Intelligence, 8,* 1–88.

Fillmore, C. (1968). The case for case. In E. Bach & R. T. Harms (Eds.), *Universals in linguistic theory* (pp. 1–88). Chicago: Holt, Rinehart & Winston.

Fillmore, C., Kay, P., & O'Connor, M. (1988). Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language, 64,* 510–538.

Gazdar, G., & Mellish, C. (1989). *Natural language processing in LISP.* Reading, MA: Addison-Wesley.

Gentner, D. (1983). Structure mapping: A theoretical framework for analogy. *Cognitive Science, 7,* 155–170.

Gentner, D., Falkenhainer, B., & Skorstad, J. (1988). Viewing metaphor as analogy. In D. Helman (Ed.), *Analogical reasoning* (pp. 171–177). Norwell, MA: Kluwer Academic.

Gibbs, R. (1989). Understanding and literal meaning. *Cognitive Science,* 13, 243–251.

Gibbs, R., & Nayak, N. (1989). Psycholinguistic studies on the syntactic behavior of idioms. *Cognitive Psychology, 21* (13), 100–138.

Hayes, E., & Bayer, S. (1991). Metaphoric generalization through sort coercion. In *Proceedings of the 29th Meeting of the Association for Computational Linguistics* (pp. 222–228). Morristown, NJ: Association for Computational Linguistics.

Hinkelman, E. (1989). *Linguistic and pragmatic constraints on utterance interpretation* (Rep. No. 288). Rochester, NY: University of Rochester, Computer Science Department.

Hinkelman, E., & Allen, J. (1989). Two constraints on speech act ambiguity. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics* (pp. 212–219). Morristown, NJ: Association for Computational Linguistics.

Hirst, G. (1987). *Semantic interpretation and the resolution of ambiguity.* Cambridge, England. Cambridge University Press.

Hobbs, J. (1979). *Metaphor, metaphor schemata, and selective inferencing* (Tech. Rep. Tech. Note No. 204). San Mateo, CA: SRI.

Indurkhya, B. (1987). Approximate semantic transference: A computational theory of metaphors and analogy. *Cognitive Science, 11,* 445–480.

Indurkhya, B. (1992). *Metaphor and Cognition.* Dordrecht, The Netherlands: Kluwer.

Iverson, E., & Helmreich, S. (1992). Metallel: An integrated approach to non-literal phrase interpretation. *Computational Intelligence, 8,* 477–493.

Jacobs, P. S. (1987). Knowledge-intensive natural language generation. *Artificial Intelligence, 33,* 325–378.

Jones, M. A., & McCoy, K. (1992). Transparently-motivated metaphor generation. In R. Dale, E. Hovy, D. Rösner, & O. Stock (Eds.), *Lecture notes in artificial intelligence: Vol. 587. Aspects of automated natural language generation* (pp. 183–198). Berlin: Springer-Verlag.

Jurafsky, D. (1993). *A cognitive model of sentence interpretation: The construction grammar approach* (Tech. Rep. No. TR 93–077). Berkeley, CA: Jurafsky.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by.* Chicago: University of Chicago Press.

Lytinen, S., Burridge, R., & Kirtner, J. D. (1992). The role of literal meaning in the comprehension of non-literal constructions. *Computational Intelligence, 8,* 416–432.

MacWhinney, B. (1991). *The CHILDES Project: Tools for analyzing talk.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Martin, C. (1993). *Direct memory access parsing* (Tech. Rep. CS 93–07). Chicago: University of Chicago, Department of Computer Science.

Martin, J. H. (1990). *A computational model of metaphor interpretation.* Cambridge, MA: Academic.

Martin, J. H. (1992). Computer understanding of conventional metaphoric language. *Cognitive Science, 16,* 233–270.

Martin, J. H. (1994). Metabank: A knowledge-base of metaphoric language conventions. *Computational Intelligence, 10,* 134–149.

Mayfield, J. (1992). Controlling inference in plan recognition. *User Modeling and User Adapted Interaction, 2,* 55–82.

Norvig, P. (1987). *A unified theory of inference for text understanding* (Rep. No. UCB/CSD 87–339). Berkeley: University of California, Computer Science Department.

Norvig, P. (1989). Marker passing as a weak method for text inferencing. *Cognitive Science, 13,* 569–620.

Pustejovsky, J. (1991). The generative lexicon. *Computational Linguistics, 17,* 409–441.

Pustejovsky, J., & Bergler, S. (Eds.). (1992). *Lecture notes in artificial intelligence: Vol. 627. Lexical semantics and knowledge representation.* Berlin: Springer-Verlag.

Russell, S. W. (1976). Computer understanding of metaphorically used verbs. *American Journal of Computational Linguistics, 2,* 15–28.

Stock, O. (1989). Parsing with flexibility, dynamic strategies and idioms in mind. *Computational Linguistics, 15,* 1–18.

van der Linden, E.-J. (1992). Idioms, non-literal language and knowledge representation. *Computational Intelligence, 8,* 433–455.

Veale, T., & Keane, M. T. (1992). Conceptual scaffolding: A spatially founded meaning representation for metaphor comprehension. *Computational Intelligence, 8,* 494–519.

Weber, S. (1994). A structured connectionist model of figurative adjective–noun combinations. In J. Barnden & K. Holyoak (Eds.), *Analogy, metaphor and reminding* (pp. 45–59). Norwood, NJ: Ablex.

Weiner, J. E. (1984). A knowledge representation approach to understanding metaphors. *Computational Linguistics, 10,* 1–14.

Wilensky, R., & Arens, Y. (1980). Phran: A knowledge-based natural language understander. In *Proceedings of the Association for Computational Linguistics* (pp. 117–121). Philadelphia.

Wilks, Y. (1975). Preference semantics. In E. Keenan (Ed.), *The formal semantics of natural language* (pp. 329–348). Cambridge, England: Cambridge University Press.

Wilks, Y. (1978). Making preferences more active. *Artificial Intelligence, 11,* 197–223.

Zernik, U. (1987). *Strategies in language acquisition: Learning phrases from examples in context* (UCLA–TR No. UCLA–AI–87–1). Los Angeles: University of California.

Zernik, U., & Dyer, M. (1987). The self-extending phrasal lexicon. *Computational Linguistics, 13,* 308–327.