MetaBank: A Knowledge-Base of Metaphoric Language Conventions

James H. Martin
Computer Science Department and
Institute of Cognitive Science
University of Colorado,
Boulder, CO
80309-0430
303-492-3552
martin@cs.colorado.edu *

CU-CS-526-91

*I would like to thank Keith Vander Linden and Zhi Hua Long for their help with the data analysis in this article. This work was supported by the National Science Foundation under Contract No. IRI-9109859.

Abstract

The frequent and conventional use of non–literal language has been a major stumbling block for natural language processing systems since the early machine translation efforts. Metaphor, metonymy and indirect speech acts are among the most troublesome phenomena. Recent computational efforts addressing these problems have taken an approach that emphasizes the use of systematic knowledge about non–literal language conventions. We are currently engaged in an effort to supply this knowledge in the case of conventional metaphor. We are constructing *MetaBank: an empirically derived and theoretically motivated knowledge–base of English metaphorical conventions.* This article describes our three–part approach to the construction of MetaBank: the collection of on–line textual resources and databases of linguistic generalizations, the development of a methodology for analyzing these resources, and the construction of a knowledge–base based on the preceding analyses.

1 Introduction

The frequent and conventional use of non–literal language has been a major stumbling block for natural language processing systems since the early machine translation efforts. Metaphor, metonymy and indirect speech acts are among the most troublesome phenomena. This is both because they occur quite frequently and because they have resisted computational efforts to deal with them in a uniform and tractable manner.

Consider the following examples of conventional metaphor.

(1) It *came* to me that I had to prepare a talk for the conference.
(2) It *hit* me that I didn't have anything to say.
(3) It *struck* me that this wasn't a good situation.

The italicized words in these examples are being used in ways that clearly deviate from the physical or spatial meaning of the words. Nevertheless, there is little that is novel or unconventional in these examples. In order to understand and generate uses like these, we make use of a rich set of underlying metaphorical conventions. In these examples, these conventions structure beliefs and believers as objects with locations. Correspondingly, change of location indicates a change in belief, as in (1) where the new idea "comes to" the believer.

The main thrust of this *Metaphoric Knowledge* approach to metaphor is that the interpretation of metaphoric language should proceed through the direct application of specific knowledge about the metaphors in the language. This approach has been embodied in MIDAS (Metaphor Interpretation, Denotation, and Acquisition System)(Martin, 1990). MIDAS is a set of computer programs that can be used to perform the following tasks: explicitly represent knowledge about conventional metaphors, apply this knowledge to interpret metaphoric language, and learn new metaphors as they are encountered.

While the MIDAS project demonstrated some significant results, it nevertheless had a major shortcoming. The effectiveness of the approach for language interpretation, generation and acquisition was obviously dependent on the size and the correctness of the knowledge–base of non-literal conventions. Unfortunately, the knowledge–base used by MIDAS does not have any kind of real coverage, nor does it have an empirically verifiable basis.

This article introduces our efforts to overcome these problems for the case of conventional metaphor. We are creating *MetaBank: an empirically derived and theoretically motivated knowledge–base of English metaphorical conventions.*

The bulk of this article provides motivation for and descriptions of various aspects of the MetaBank project. Section 2 presents a brief review of knowledge–based linguistic approaches to processing non–literal language. Section 3 reviews the current problems with this approach. Section 4 briefly surveys the three main components of the project: the collection of linguistic resources, our methods of metaphor analysis, and the knowledge–base itself.

The remainder of the article is devoted to the results of our initial corpus–based metaphor analysis. With respect to this corpus, we have focussed on answering the following questions:

- How frequently does metaphor occur?
- What metaphors are there?
- What is the relative frequency of individual metaphors?
- How many metaphors are there?

Section 6 presents our answers to these questions.

2 Knowledge-Based Linguistic Approaches

MIDAS can be seen as a part of a recent trend in the area of semantic interpretation that uses systematic knowledge about non–literal language conventions. Among the research projects following this general approach are Zernik's RINA system (phrasal idioms) (Zernik, 1987), Fass's META5 system (metaphor and metonymy) (Fass, 1988; Fass, 1991), and Hinkelman's system (indirect speech acts) (Hinkelman, 1989; Hinkelman and Allen, 1989). These systems all attempt to leverage knowledge of systematic language conventions in an attempt to avoid resorting to more computationally expensive methods. At the same time, however, these approaches have all avoided simplistic phrasal or word-sense approaches that neither adequately capture generalizations nor allow for novel cases. Finally, like MIDAS, they all use small knowledge–bases of conventions with minimal coverage and little empirical validation. (Hinkleman's corpora-based (Hinkelman, 1990) research is a notable exception to this). The following discussion presents an overview of how this general approach has been applied to the problem of conventional metaphor.

Under this knowledge–based approach, the way to approach the study of metaphor is to study the underlying details of individual metaphors, and systems of metaphors in the language. This approach follows on the metaphor work of Lakoff and Johnson (Lakoff and Johnson, 1980) and builds directly on the knowledge–based computational approaches to metaphor of Jacobs and Norvig (Jacobs, 1985; Norvig, 1987). Earlier work on metaphor by Wilks and Hobbs (Wilks, 1978; Hobbs, 1979) are the most relevant computational predecessors to this work.

2.1 MIDAS

Since the remainder of this paper focuses on our MetaBank work, this section will first briefly illustrate how such a knowledge–base has been used. This is not intended as an exhaustive list since the knowledge–base certainly will have many unanticipated uses. Specifically, this section describes how the MIDAS system, as a part of the UNIX Consultant (UC) system , made practical computational use of the metaphors stored in its knowledge–base.

In particular, it introduces the following three issues.

Representation: The explicit representation of the conventional metaphors in a language in the form of explicit associations between concepts.

Interpretation: The correct and efficient application of this metaphoric knowledge to the interpretation of metaphoric language.

Learning: The dynamic acquisition of new knowledge about metaphors for which no known metaphor provides a coherent explanation.

Earlier descriptions of MIDAS (Martin, 1990; Martin, 1992) give more complete details on these topics.

2.2 Knowledge Representation

Consider the following simple example of a conventional UNIX metaphor. The metaphorical use of the word *in* reflects a systematic metaphorical structuring of Unix processes as enclosures.

(6) I am *in* Emacs.

Metaphors like this may be said to consist of the following component concepts: a *source* component, a *target* component, and a set of conventional associations from the source to target. The target consists of the concepts to which the words are actually referring. The source refers to the concepts in terms of which the intended target concepts are being viewed. In this example, the target concepts are those representing the state of currently using a computer process. The source concepts are those that involve the state of being contained within some enclosure.

The approach taken here is to explicitly represent conventional metaphors as sets of associations between source and target concepts. The metaphor specifies how the source concepts reflected in the surface language correspond to various target concepts. In this case, the metaphor consists of component associations that specify that the state of being enclosed represents the idea of currently using the editor, where the user plays the role of the enclosed thing, and the Emacs process plays the role of the enclosure. Note that these source-target associations are represented at the conceptual and not the lexical level. Any single lexical item or expression that can be construed as referring the source concept of a known metaphor may invoke that metaphor. In this example, the source component of the metaphor is attached to the concept of being enclosed, not to the lexical item *in*.

These sets of metaphoric associations, along with the concepts that comprise the source and target domains, were represented using the KODIAK (Wilensky, 1986) representation language. KODIAK is an extended semantic network language in the tradition of KL-ONE (Brachman and Schmolze, 1985) and its variants. These sets of metaphoric associations are full-fledged KODIAK concepts. As such, they can be related to other concepts and arranged in abstraction hierarchies using the inheritance mechanisms provided by KODIAK. Specifically, KODIAK was used to represent specialized domain specific metaphors, pervasive high-level metaphors, and the systems of relations among related metaphors. Section 4.3 will discuss the representation being used in the current MetaBank work.

2.3 Interpretation

The interpretation process in MIDAS is basically one that views a given input sentence as providing a set of constraints on possible interpretations. MIDAS checks the input constraints against all the possible interpretations that can be conventionally associated with the input. Interpretations that are coherent with the constraints are returned. The possible conventional associations include direct non-metaphoric interpretations, as well as all the conventional metaphors that are invoked by the input.

Consider briefly the details of the following shortened trace of a Unix example. In this example, UC calls upon MIDAS to find a coherent interpretation for this use of *enter*. MIDAS finds, and attempts to apply, all the conventional metaphorical and non-metaphorical concepts associated directly with, or inherited by, this concept. In this case, it finds that the only conventional interpretation that is consistent with the input is the one that results from the application of the known Enter-Lisp metaphor.

```
> (do-sentence)
Interpreting sentence:

How can I enter lisp?

Interpreting concreted input.

(A Entering50 (↑ Entering)
   (enterer50 (↑ enterer) (A I203 (↑ I)))
   (entered50 (↑ entered) (A Lisp58 (↑ Lisp))))

Failed interpretation:  Entering50 as Entering.
Failed interpretation:  Entering50 as Enter-Association.
Valid known metaphorical interpretation:  Entering50 as Enter-Lisp.
```

The parser first produces a syntactic analysis and a preliminary semantic representation. At this point in the analysis, UC calls upon MIDAS to begin a deeper analysis of this initial representation.

The case structure of this preliminary representation is then checked against the semantic constraints of the interpretations conventionally associated with the Entering concept. In this case, MIDAS finds that the direct interpretation and one of the other known Entering metaphors can be rejected before the appropriate Enter-Lisp metaphor is found.

```
Applying conventional metaphor Enter-Lisp.

(A Enter-Lisp  (↑ Container-Metaphor Metaphor-Schema)
   (enter-lisp-res enter-res → lisp-invoke-result)
   (lisp-enterer enterer → lisp-invoker)
   (entered-lisp entered → lisp-invoked)
```

```
  (enter-lisp-map Entering → Invoke-Lisp))

Mapping input concept Entering50 to concept Invoke-Lisp30
  Mapping input role enterer50 with filler I203 to
               target role lisp-invoker30
  Mapping input role entered50 with filler Lisp58 to
               target role lisp-invoked30

Yielding interpretation:

(A Invoke-Lisp30 (↑ Invoke-Lisp)
   (lisp-invoked30 (↑ lisp-invoked)
    (A Lisp58 (↑ Lisp)))
   (lisp-invoker30 (↑ lisp-invoker)
    (A I203 (↑ I))))
```

MIDAS then begins the process of mapping from the given source concepts to the appropriate target concepts based on the constraints imposed by the metaphor. The mapping process, called *metaphoric unviewing*, creates a new instance of the target concept for use in further processing. Any further inferences that need to be performed are based on this newly created target concept. In this example, the source concept of `Entering` is mapped to the target concept `Invoke-Lisp` as specified by the metaphor.

```
Calling UC on input:

(A How-Q207 (↑ How-Q)
   (topic206 (↑ topic)
    (A Invoke-Lisp30 (↑ Invoke-Lisp)
       (lisp-invoked30 (↑ lisp-invoked) (A Lisp58 (↑ Lisp)))
       (lisp-invoker30 (↑ lisp-invoker) (A I203 (↑ I))))))

UC: You can enter lisp by typing 'lisp' to the shell.
```

Finally, UC uses this new target concept as the basis for answering the user's question by using its long-term knowledge about how to initiate the Lisp system. Note that UC makes use of the metaphor in expressing its answer to the user.

2.4 Learning

MIDAS will inevitably face the situation where a metaphor is encountered for which none of its known metaphors provides an adequate explanation. This situation may result from the existence of a gap in the system's knowledge-base of conventional metaphors, or from an encounter with a novel metaphor. In either case, the system must be prepared to handle the situation. Consider the following example.

In this example, the user has employed the conventional UNIX metaphor that the termination of an ongoing process can be viewed as a killing. However, unlike the previous example, MIDAS finds that it is initially unable to interpret this example because it has no knowledge of this conventional metaphor. More precisely, it determines that the given input can not adequately satisfy the constraints associated with any of the concepts conventionally associated with the word *kill*.

```
> (do-sentence)
Interpreting sentence:

How can I kill a process?

Interpreting concreted input.

(A Killing16 (↑ Killing)
   (killer16 (↑ killer) (A I46 (↑ I)))
   (kill-victim16 (↑ kill-victim)
    (A Computer-Process10 (↑ Computer-Process))))

Failed interpretation:  Killing16 as Killing.
Failed interpretation:  Killing16 as Kill-Delete-Line.
Failed interpretation:  Killing16 as Kill-Sports-Defeat.
Failed interpretation:  Killing16 as Kill-Conversation.

No valid interpretations.
```

At this point, MIDAS has exhausted all the possible conventional interpretations of the primal representation. In particular, the direct non-metaphoric interpretation and three known metaphorical interpretations are rejected because their restrictions of the role of the `kill-victim` fail to match the semantics of the concept filling that role in the input, a `computer-process`.

```
==============================================================
Entering Metaphor Extension System
==============================================================

Attempting to extend existing metaphor.

Selecting metaphor Kill-Conversation to extend.

Attempting a similarity extension inference.

Creating new metaphor:  Killing-Terminate-Computer-Process

(A Killing-Terminate-Computer-Process (↑ Kill-Metaphor)
   (kill-victim-c-proc-termed-map kill-victim → c-proc-termed)
   (killer-c-proc-termer-map killer → c-proc-termer)
   (killing-terminate-computer-process-map Killing
    → Terminate-Computer-Process))
```

This example illustrates the operation of the learning component of MIDAS, the Metaphor Extension System (MES). This system is invoked by MIDAS when it discovers a metaphor for which it has no adequate knowledge. The task of the MES is to attempt to extend its knowledge of some existing metaphor in a way that will yield a coherent interpretation for the new use and provide a basis for directly understanding similar uses in future. In this case, the system finds and extends a known closely related metaphor that also uses *kill* to mean terminate.

```
Final interpretation of input:

(A How-Q46 (↑ How-Q)
   (topic46 (↑ topic)
    (A Terminate-Computer-Process10
       (↑ Terminate-Computer-Process)
       (c-proc-termer10 (↑ c-proc-termer)
(A I46 (↑ I)))
       (c-proc-termed10 (↑ c-proc-termed)
(A Computer-Process10
   (↑ Computer-Process))))))

UC: You can  kill a computer process by typing ^c to the shell.
```

Finally, the target concept determined by the MES is used to provide an answer to the user.

The approach taken in MIDAS to the understanding of new or unknown metaphors is called the Metaphor Extension Approach. The basic thrust of this approach is that a new metaphor can best be understood by extending an existing well-understood metaphor or combining several known metaphors in a systematic fashion. Under this approach, the ability to understand and learn new metaphors depends critically on the availability of systematic knowledge about existing known metaphors. This approach to learning can be seen as an example of *Analogical Reasoning* (Helman, 1988; Prieditis, 1987) or *Case-Based Reasoning* (Hammond, 1986; Kolodner and Simpson, 1989).

3 Limitations to the Knowledge–Based Paradigm

As mentioned in section 1 the primary current limitation to systems following a knowledge–based approach to non–literal phenomena is the size and correctness of the knowledge–base itself. Consider the ways that this general problem manifests itself in MIDAS.

- There is no way of telling if the known metaphors that are included are represented correctly.
- There is no way of telling if a given metaphor is of high enough frequency and range to warrant significant effort.
- MIDAS lacks any empirical basis for both the abstract domain independent metaphors, and the domain specific UNIX metaphors, that are included in its knowledge-base.
- There is no way of determining the coverage of the current knowledge–base. How many conventional metaphors are really out there?
- There is no empirical basis for the particular hierarchical relationships embodied in the current knowledge–base.

Of course, these problems of knowledge–base size, correctness and organization do not differentiate these natural language processing efforts from many other knowledge–based AI efforts. However, within the field of natural language processing, significant progress has recently been made in the construction of large lexicons and larger more realistic grammars. The research we are engaged in continues this trend.

4 The MetaBank Project

The MetaBank project is an attempt to solve some of the above problems for the case of conventional metaphor. To be specific, we are identifying resources and developing methodologies that will allow us to construct a robust knowledge–base of English metaphoric and metonymic conventions. This knowledge–base is being represented in a form that will make it generally useful to natural language processing applications.

We are taking a three–part approach to the construction of MetaBank. The first part involves the collection of on–line textual resources and databases of linguistic generalizations. The second part is the development of a methodology for analyzing these resources and deriving generalizations from them. The final part is the actual construction and use of a knowledge–base based on the preceding analyses. The following sections will briefly introduce these three parts.

4.1 Resources

We are making use of the following on–line resources as a part of this effort.

1. Berkeley Metaphor List: An on–line list of known English metaphors with analyses.
2. UNIX Texts: a corpus user mail dealing with the UNIX operating system.
3. Wall Street Journal: several years worth of the Wall Street Journal as provided by the Association for Computational Linguistics Data Collection Initiative. [1]

4.1.1 Berkeley Metaphor List

The Metaphor Project, at the University of California at Berkeley, under the direction of George Lakoff, has been collecting an on-line database of conventional English metaphors. The database entries for each metaphor include descriptions of the source and target domains, a set of example sentences using the metaphor, a short analysis, and where appropriate, pointers to a longer analysis from which the entry was derived. The entries are being compiled by hand from the published metaphor literature, and from an on-going series of graduate seminars at Berkeley. Currently, a total of approximately 200 metaphors of varying levels of specificity have been entered into the database.

This Berkeley collection, merged with the metaphors already represented and analyzed as a part of the MIDAS project, form the starting point for MetaBank. As will be discussed below, it will also be used to guide the analysis of the various corpora.

4.1.2 UNIX Text

Our source of UNIX related text is an archive of user mail to a *trouble* alias at the University of Colorado. Mail to this alias is monitored by systems personnel and dealt with. A total of approximately 150,000 words of text (both user queries and consultants answers) have been collected. The primary practical reason for the use of text from the UNIX domain is that the primary testbed for our approach are natural language application programs that operate in this domain. What we are seeking is a means by which generic English metaphors are specialized into particular domains.

[1] The Association for Computational Linguistics has initiated the Data Collection Initiative (DCI). The purpose of DCI is to make large text corpora available as a research tool. Among other texts the initial release from DCI includes text from the 1987, 1988 and 1989 Wall Street Journal.

4.1.3 Wall Street Journal Text

The Wall Street Journal (WSJ) corpus is, of course, quite different from the our UNIX text both in size and content. Although the WSJ corpus contains articles ranging from movie reviews to reports on world events, its overwhelming focus is on economic issues and on the day to day affairs of individual businesses. The size of the corpus and the breadth of material should lead us to a wider range of metaphors than the smaller and more limited UNIX corpus.

4.2 Metaphor Analysis

By metaphor analysis we have in mind two distinct notions. The first pertains to the method of analyzing individual sentences used to identify underlying conventional metaphors. The second pertains to how we go about hunting for metaphors in a large corpus.

4.2.1 Conventional Metaphor Analysis

We are interested in the kind of common–place conventional metaphor introduced by Reddy (Reddy, 1979) and elaborated on by Lakoff and Johnson (Lakoff and Johnson, 1980) among many others. These are common–place uses of words and phrases that depart from their literal meanings in ways that systematically indicate that one concept is being thought of and spoken about in terms of another concept of a different kind. Consider the following metaphor analysis of a text from our UNIX text source.

```
>From rieman@tigger  Wed Apr 11 17:27:13 1990
Received:  by tigger.colorado.edu (cu.generic.890828)
Date:  Wed, 11 Apr 90 17:27:08 MDT
From:  John Rieman <rieman@tigger>
To:  trouble@eclipse
Subject:  locked terminal

My terminal hanging off of eclipse tty06 is locked again.    I can see
the processes (login to tigger) on eclipse, but I can't kill 'em --
eclipse says they aren't mine.

For your info here's how this happened (it has happened before):

I typed ``remote'' at the eclipse login prompt and logged into tigger.
My default terminal setting is vt100 -- that's what I use most of
the time.    I only use the zenith when I'm not doing anything
important or time-consuming.

I forgot to tell tigger I was using a zenith and I more'd a file.

The zenith locked.
```

```
I hit shift-reset to knock some sense into the zenith, then pressed
ctrl-z to get out of more.   I killed the more process.

I typed ``setenv TERM z89'' to tell tigger I was using a zenith.

Then I tried again to use more.   It locked again, and that's where I
am now.   Resetting the terminal has no effect, and killing my tigger
process from another machine has no effect.

-john
```

Consider some of the metaphorical conventions illustrated by this message. (Eclipse and tigger are the names of computers in the following).

`I can't` *kill* `'em -- eclipse` *says* `they aren't` *mine*

The use of *kill* here reflects a metaphor that views processes as being alive, where termination corresponds to killing. The use of *say* reflects the metaphor that views computers as communicative agents capable of carrying on dialogs with users. In this case, program output is viewed as dialog by the computer. Finally, the use of the possessive *mine* illustrates the process as possession of the user metaphor, that underlies many of the security mechanisms in computer systems.

`...ctrl-z to` *get out of* `more. I` *killed* `the more process.`

This sentence illustrates a use of the process as container metaphor. It manifests itself here in the notion of exiting or *get out of* as suspension of an ongoing process. Note that in the immediately following sentence the user switches metaphors to refer to a different aspect of the same process.

`...to` *tell* `tigger I was using a zenith...`

This sentence is another instance of the communicative agent metaphor. Here, however, we see simple input to a program as a speaking to the computer.

`...and thats` *where I am* `now...`

This final example is an illustration of a general metaphor structuring states as locations. Note that this final metaphor illustrates an important point. These domain specific corpora will also contain uses of non-domain specific metaphors. Therefore, domain independent abstract metaphors will be derived from all the textual sources we collect, including the technical ones.

4.2.2 Analyzing Texts for Metaphors

One method of analyzing our texts for metaphors would be to simply perform an exhaustive line–by–line analysis of the text. There are two problems with this approach: it makes no practical use of the wealth of information already known about English conventional metaphors, and it is impractical in the case of a large corpus like the Wall Street Journal. To overcome these problems we are employing two separate kinds of search: directed search for known metaphors, and random sampling.

4.2.3 Directed Search

The first text analysis method uses the Berkeley metaphor list as a generator of probes into the various text corpora. The following steps illustrate the method.

Step 1: Choose a metaphor of interest from the current list of metaphors. This is the target metaphor.

Step 2: Generate a set of linguistic probes for this metaphor. This involves selecting a set of words associated with the source domain of the chosen metaphor. The selection of these words will be guided by the analysis of the metaphor provided by the Berkeley list to ensure that the words are likely to occur with the metaphor. These words will typically be chosen from a well–defined and well–studied semantic field from a spatial or physical domain.

Step 3: Choose a corpus of interest appropriate to the metaphor being studied. This will typically involve an alternation between the domain specific corpora and the more general corpora. The purpose of this alternation is to determine that the behavior of the metaphor in the domain specific corpora is consistent with its behavior in other domains and with its description in the Berkeley list.

Step 4: Search the selected corpus for uses of the probe words.

Step 5: Classify the resulting sentences according to their meaning. This classification step will initially be dependent on the subjective ability of the human classifier to identify various uses. This step will result in one the following possibilities.

1. A literal use of the probe word.
2. An instance of the metaphor in question.
3. Another metaphor in the known metaphor list.
4. Another conventional metaphor not yet in the list.
5. An isolated homonymous word sense with no metaphoric basis.
6. A novel use of some kind.

The result of each probe is, therefore, a combination of information about the metaphor in question, and information about other uses of the probe words. Some of the information that can be gleaned about the target metaphor is:

- How frequently does this metaphor occur?
- How frequently are the probe words used with the target metaphor?
- How many of the probe words from the source semantic field actually appear with this metaphor?
- What is the correct level of abstraction for the source and target domains of the metaphor?

Some of the information that can be gotten from negative probe sentences (sentences not relevant to the target metaphor) is:

- Identification of conventional metaphors not in the current database.
- Frequency information about other metaphors.
- Frequency information about the probe word with other meanings.
- Identification of novel metaphors. (Metaphors that would not be judged as conventional).
- Frequency of occurrence of novel uses.

Note that this method of classifying the use of probe words is analogous to word-sense tagging (Zernik, 1991; McRoy, 1992). In these efforts, various kinds of local context is used to classify a use of a word in a text as an instance of a pre-determined dictionary sense. In our work, the role of the dictionary sense is replaced by a conventional metaphor. Also relevant is the corpus work done by Hinkelman in studying speech acts. In that work, words such as *please* and *hereby*, which can be used as performatives, were used as probes into a large corpus. The resulting text was then analyzed to determine the intended meaning of each occurrence.

This methodology can be seen as a combination of top-down and bottom-up techniques. On the top–down side it makes full use of the analytical linguistic work that has been done (as it is represented in the Berkeley database) to guide the search for metaphors in a text. The bottom-up side is the actual examination of large amounts of text. Note that the methodology we are using allows us to probe the corpus in a focussed manner by using metaphors from the Berkeley list, while at the same time it allows us to discover metaphors not contained in the original list. Section 5 presents the results from a directed search in the UNIX corpus for container metaphors.

4.2.4 Sampling

While the directed search strategy gives detailed information about the frequency and nature of selected metaphors, it says little about the overall picture of the nature and frequency of the metaphors in a corpus. In particular, it is a poor way to capture as yet unknown conventional metaphors.

The most straightforward way of acquiring such information would be to do an exhaustive metaphor analysis of each sentence in the corpus. Indeed such exhaustive analyses have been performed for the purpose of determining the overall frequency of metaphor and metonymy in a wide variety of corpora (Pollio et al., 1990). These studies have found that these phenomena appear quite frequently, averaging around 5 uses per 100 words of text. Unfortunately these studies do not report what types of metaphors were found nor their individual frequencies.

While an exhaustive sentence by sentence study of the WSJ corpus to determine this information would be useful, it was clearly impractical for us given the size of the corpus. A more reasonable method might be to use an automated metaphor analysis system. Unfortunately, there are currently no robust automated metaphor tools analogous to part–of–speech taggers, word sense taggers, or parsers that would reliably permit large–scale automatic analysis.

Fortunately, two considerations led us to believe that a simple sampling method would prove useful. The first consideration stems from the frequency results cited above that indicate that conventional metaphors of the type we are interested in are quite frequent. The second consideration is a fundamental assumption of all work on conventional metaphor. This is that there is a relatively small core set of important conceptual metaphors. Taken together these considerations imply that an analysis of a relatively small number of sentences randomly sampled from the corpus would provide useful information. Section 6 presents the results of such an analysis.

4.3 Building the Knowledge-Base

The first phase of construction involves integrating all of the metaphors from the Berkeley list into the current MIDAS knowledge–base. This involves representing the following kinds of knowledge.

- Knowledge about the source concepts of the metaphor. Particular care will be taken with the representation of the source concepts since it is assumed that they will occur across domains.
- Knowledge of the target concepts.
- Knowledge about the metaphors themselves. This information will largely come from the analyses of the metaphors from the Berkeley list.

The representation of this knowledge will entail significant effort since the Berkeley list does not give detailed information about the source and target domains. In this phase, we intend to make extensive use of existing formalizations of physical and spatial domains (Hobbs et al., 1987; Lenat and Guha, 1990).

As noted above, the Berkeley list currently contains approximately 200 English metaphors. The current MIDAS knowledge-base contains 22 domain independent metaphors (11 of which are contained in the Berkeley list) and an additional 18 UNIX specific specializations. We believe that a stable MetaBank for English will contain no more than 500 high level generic metaphors.

As noted above, the knowledge in the original MIDAS knowledge–base was represented using the KODIAK representation language. Briefly, MIDAS currently uses a hierarchical knowledge–base of structured associations linking source and target concepts that are conventionally related metaphorically. Although KODIAK is similar to other extended semantic network formalisms, it is not in widespread use and therefore would not be an appropriate choice for a knowledge–base intended for distribution.

We have instead decided to use the Ontolingua knowledge representation language (Gruber, 1992). Ontolingua is intended to be used to represent and distribute common ontologies. It supports a canonical format which can be translated into a number of common representation systems including LOOM (Brill, 1992). Facilities are also provided for the creation of generators for other representation systems like KODIAK. Ontolingua is based on the KIF knowledge interchange format (Genesereth and Fikes, 1992). It is an attractive representation system for MetaBank because it allows us to represent metaphors in a single, canonical form that can be translated into the disparate representation systems used by potential MetaBank consumers.

5 A Directed Search Example: The Container Metaphor

Consider the following directed search example. In this example, the Boulder `trouble` mail archive is probed for instances of container metaphors. This is one of the more productive and well-studied kinds of metaphor in English. The Berkeley Metaphor list contains thirteen container metaphors (thirteen distinct target domains structured with the source concept container). In addition, significant effort was spent on computer system container metaphors as a part of the MIDAS project.

For the purposes of the this example, the probes *enter, get into, exit* and *get out of* from the source domain of containers or enclosures were used to probe the selected corpus. Table 1 shows the results.

choose "search" from the menu	enter "X", hit "apropos", and when the
BTW, when I try to login, it let's me	enter my password, then prints daveheib
Randomly in these two windows when I	entered the vi editor
I	enter my password when 'rlogin'g in from
prompt I have to be right there to	enter my passwd
Kermit tells me to	enter a Receive command and I do so.
I have a file called calendar that I	enter appointments
in. It seems that whenever I	enter an item in this file,
the appointment that I just	entered. It's not a huge
As soon as I	enter windows it fails (most of the time).
they	get into the queue, but they don't print
Oh, good grief! I can't	get into my home directory
log me into my home directory and I cannot	get into it by any other means.
a program that attempts to	get into computers around the Internet.
an intruder making persistent attempts to	get into Internet
I can then do a successfull :vi to	get into vi
hung up when I tried	exiting out of X windows. Keyboard, mouse, etc.
I tried to	exit, at which point I received the prompt
I was	exiting suntools when cashew choked in a
now hangs when I want to	exit and save using the VI editor
window, log into the machine. Now,	exit the login
i was on the console	exiting suntools with crtl-d crtl-q
pressed ctrl-z to	get out of more.
the only way I can	get out of it is to completely shut
would like to get my degree and	get out of here sometime

Table 1: Container Metaphor Examples

This probe yielded the following results:

1. All the probes appeared with the interactive–system as enclosure metaphor that was extensively studied with MIDAS. This was also the most frequent use.
2. A use of the metaphor with the MORE command which neither UC nor MIDAS had classified as an interactive–system.
3. A conventional use, structuring the home directory as an enclosure for the user, that was not known to MIDAS. It was used only with *get into*.
4. A conventional use of *enter* as in *to enter your passwd.* Again not known to MIDAS, and not discussed in the Berkeley list.
5. A relatively new use structuring programs as entering computers from the outside. As in *a program that attempts to get into computers around the Internet.* Mentioned in the Berkeley list, not known to MIDAS.
6. One generic use contained in the Berkeley list, *get my degree and get out of here*, that could be considered literally as leave, or metaphorically as graduate.

The net result of this was a strong confirmation of the Interactive-Process-As-Enclosure metaphor as currently known to MIDAS. It occurred quite frequently, with the characteristics predicted by the current representation. It also resulted in two conventional uses that have to be added to the metaphor list and added to MIDAS, and one new computer use that also needs to be added.

Another important result of this probe is illustrated by the existence of multiple container metaphors structuring different aspects of the same target domain. This result indicates that abstract high level metaphor schemas, while useful, are not sufficient to uniquely predict the particular instantiations that show up in specific domains.

6 A Wall Street Journal Sampling Experiment

Following the sampling strategy outlined above we conducted the following experiment using the WSJ corpus. Six samples of approximately 100 sentences each were randomly selected from the corpus. Each sampled sentence was then was analyzed for conventional metaphors. Each sentence containing any kind of metaphor, conventional or novel, was retained. The resulting sentences were then classified in terms of their source and target concepts using an analysis explained below.

The resulting analysis allowed us to produce preliminary answers to the following questions:

- How frequent is metaphor in this corpus?
- What metaphors occur in this corpus?

- What is the relative frequency of these metaphors?
- How many metaphors are there?

6.1 Overall Frequency of Metaphor

Our sampling results are largely in agreement with the frequency results previously obtained through exhaustive analyses of a variety of English (Pollio et al., 1990) texts. Our sample produced 515 instances of metaphors out of 581 sampled sentences containing 11925 words of text. This yields a frequency of 4.3 metaphors per 100 words of text.

6.2 What Metaphors Are There?

The real issue here concerns the number of conventional conceptual metaphors underlying the metaphor instances found in the sample. To answer this question the metaphor instances were analyzed and grouped into related clusters. Instances were clustered together whenever they are judged to be closely related conceptually at some meaningful level of abstraction in both their source and target components. This clustering of the 515 metaphor instances resulted in 102 coherent metaphor clusters. Table 2 shows the five largest clusters from this experiment. The name of each cluster is followed by the number of instances that it contains and an example sentence.

Value-As-Location/Value-Change-As-Movement: 132

> Spain Fund *tumbled* 23% in *turbulent* trading, *dragging down* the shares of other so-called country funds.

Group-as-Person/Animate: 43

> But if *it changed its mind*, however, it would do so for investment reasons, the filing said. (*The "it" in this sentence was referring to a corporation.*)

Abstract-Indicators-As-Sensory-Information: 34

> The world market for physical sugar was *quiet* and there was a lack of any other fundamental developments, analysts said.

Activities/Events-As-Paths: 29

> So where does the IMF *go from here*?

Organization-as-Structured-Object: 25

> The entire single-country fund *component* of the industry is being re-rated to much higher valuations, said Thomas J. Herzfeld, who runs a Miami investment management firm that bears his name.

Table 2: Top Five Metaphor Clusters

To illustrate the nature of these clusters consider the following examples from the Value-As-Location/Value-Change-As-Movement cluster.

> The Financial Times 30-share index *tumbled* 34.9 to 1822.9.
>
> Blue chips *cushioning the fall* in the Dow industrials included Boeing, which gained 1 1/8 to 59 1/2 as investors anticipated a possible $5 billion plane order from Northwest Airlines and an expected resumption of contract talks with striking machinists.
>
> While net income doubled to an estimated $9.2 million during the nine months ended July 31, profit margins *fell* to 3.9% from 5.8% a year earlier, as general expenses *soared* nearly sevenfold.
>
> Over the course of the latest recession of 1981-82, service-industry employment *rose* about 200,000, against a 2.7 million *drop* in goods-producing jobs; even so, overall unemployment *soared* to nearly 11% from about 7%.

The terms in italics from the source domain, while they differ in details, all involve motion involving a change in location. Hence they all satisfy the criteria of conceptual relatedness for the source. The target concepts also differ widely in details. The targets include stock prices, employment rates, profit margins, and expenses. What they all have in common is that they can be characterized by a single feature having a numerical value. At that level of abstraction they are related and hence are all included in this cluster.

It should be noted that this clustering produces extremely coarse results. The characterizations of the resulting clusters, while useful, are not sufficiently detailed to serve either as entries in the Berkeley Metaphor List or the MIDAS knowledge–base. In particular, they do not capture the internal hierarchical structure that has been systematically observed in conventional metaphor. For example, the above Value-Change-As-Movement metaphor clearly divides into two distinct subclasses: motion up and down, and motion forward and backwards. The following example illustrates how these two distinct metaphors can co–exist in the same context with no systematic conflict.

> The dollar *shot higher* while stocks and bonds *retreated* after traders interpreted remarks by Federal Reserve Chairman Alan Greenspan to mean the central bank won't *lower* interest rates to halt the dollar's rally.

For the purposes of this experiment it was decided that the target component of this cluster would be represented as change of location rather than dividing it into two sub-metaphors. However, when this metaphor is included into the MetaBank the finer grained distinctions will have to be represented.

An overall examination of the source domains from all the clusters reveals many of the usual conceptual domains cited in the linguistics literature (Talmy, 1975; Talmy, 1988; Lakoff, 1987; Jackendoff, 1983) as important conceptual structuring devices. The following are among the more frequent source domains.

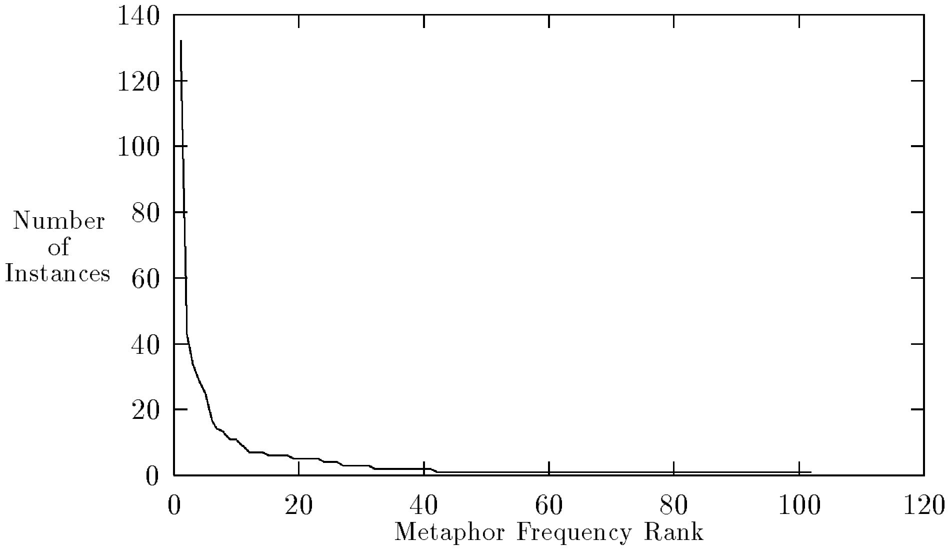

Figure 1: Rank Frequency Plot of Metaphor Clusters

Topology: locations, orientations, motion, paths, containment

Sensory: visual, auditory, olfactory, tactile, temperature

Living things: animacy, body–part relations, reproduction, health

The target domains of these metaphor clusters largely reflect the core conceptual content of this corpus, thus reinforcing the view that metaphor is a device that addresses core conceptual content of a domain, not peripheral issues.

6.3 Relative Frequency

As is hinted at in Table 2, the frequency distribution of the metaphors clusters is far from uniform. The top five metaphor clusters account for over half of all the metaphor instances. At the other end of the scale, 41 of the 102 metaphor clusters appear only once. Figure 1 shows the overall rank frequency distribution of the clusters.

This plot shows that a small number of metaphors account for a large number of instances, with the curve tailing off significantly to extremely low frequency metaphors. It appears, therefore, that the conventional metaphors in this corpus closely follow the Zipf's Law (Zipf, 1949) behavior observed in so many domains. This Zipf-like behavior has potential implications for both computational and psychological processing models of metaphor understanding. The pattern of memory access for conventional metaphors or for source analogs is far from uniform.

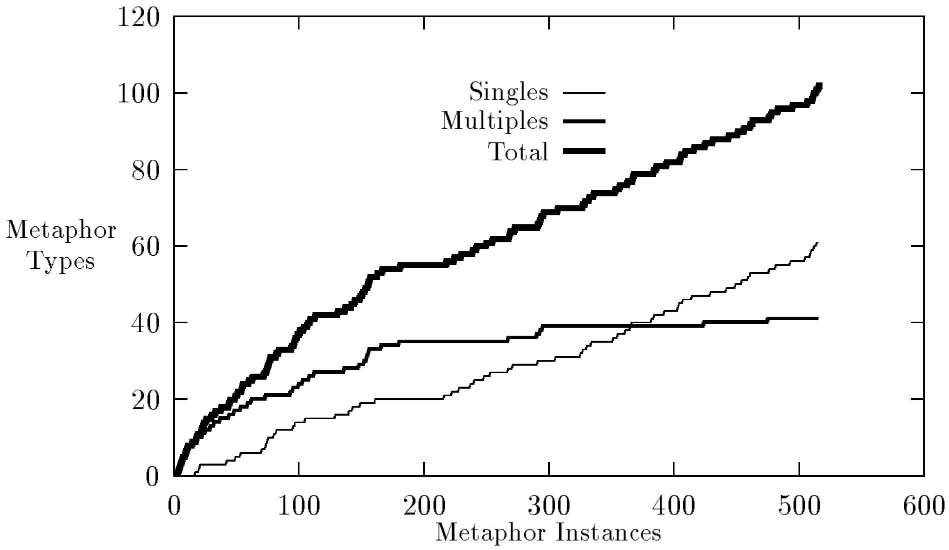

Figure 2: Cumulative Metaphor Growth

6.4 How Many Metaphors Are There?

The answer to this question is critical to the overall goals of the MetaBank project. A fundamental assumption in this work is that the number of core conceptual metaphors that can cover most metaphor instances is relatively small, making the construction of the knowledge–base a reasonably tractable task. Any answer given here is somewhat speculative given the small fraction of the WSJ corpus that we have analyzed, and the specialized nature of the corpus. However, the relative frequency information does give an indication that some useful conclusions can be drawn.

Consider the plots shown in Figure 2. The thickest line plots the growth in the cumulative number of metaphor clusters as a function of the number of instances seen. This curve rises sharply at first then slows slightly but nevertheless shows no sign of tailing off. This is a hardly an encouraging prospect considering the size of the WSJ corpus. It is possible that this cumulative curve would level off given a larger sample size, but with the current sample its impossible to tell.

Fortunately, the uneven relative frequency of the metaphors offers some insight into this growth curve. The thinner lines in Figure 2 show the cumulative growth in the number of metaphors separated into metaphor clusters that occur once and those that occur more than once.

The growth of the thinnest metaphor curve reflects the fact that there is a class of metaphors that occurs with such low frequency that they are effectively seen once and never seen again. This type of single instance metaphor does nevertheless occur at a constant rate, mirroring the apparently linear growth in the cumulative number of metaphors. This result can be attributed to the occurrence of novel metaphors and extremely low frequency conventional metaphors.

The shape of the multiple instance metaphor curve is much more promising. The major growth in multiple instance metaphors appears very early in the sample and then rapidly levels off around 42. Moreover, the five most frequent clusters listed above in Table 2, which account for more than half the metaphor instances, were all encountered within the first 28 sampled sentences.

These results indicate that it should in principle be possible to identify the core metaphors in a domain rapidly without having to analyze a significant portion of a corpus. Nevertheless, despite the obvious trends in Figure 2, the small size of our sample renders the question of how many core metaphors there are too speculative. We plan in future research with larger samples to employ a variety of population estimators to determine reasonable estimates on the number of core metaphors in English.

7 Related Areas

There are a number of on–going areas of research that are relevant to the work we are engaged in here. These include the construction of large machine usable lexicons and the construction of large common–sense knowledge-bases.

7.1 Large Lexicons

In recent years, the lexicon has taken on a more central role, both in linguistic theories, and in computational efforts. Paralleling this trend, there has been increasing interest in the construction of large, robust, on–line lexicons. The advent of machine readable dictionaries and large text corpora has made much of this work possible. The notion of a word-sense, however, remains problematic in much of this work. Many of the fine grained sense-distinctions made by lexicographers often seem unmotivated and arbitrary. On the other hand, many of the senses needed in the specialized domains in which these lexicons will be used are missing altogether.

We believe that a robust MetaBank would largely alleviate this problem with lexicons. If a word sense can safely be predicted from the spatial or physical meaning of the word combined with a known metaphor then the sense need not be listed in the lexicon.

7.2 Common–Sense Knowledge–Bases

CYC (Lenat and Guha, 1990) and Tacitus (Hobbs et al., 1987) are two major recent efforts to construct large common–sense knowledge–bases. The MetaBank project relates to these projects in a number of ways. It obviously requires a common–sense conceptual representation for the various source and target domains that play roles in conventional metaphors.

The second point of contact with these efforts is a more subtle one. Builders of common–sense knowledge–bases often find themselves in one of two problematic situations when representing various domains. The first situation happens when the designer has produced a logically consistent ontology for a domain based on non-language criteria, that bears no obvious relationship to the way that the domain is actually expressed in natural language.

Consider, as a concrete example, the various temporal logics that have been developed to reason about time. It is quite clear that none of these logics can predict the following conventional language uses.

(4) I had a lot of time to *kill*.
(5) I can't *waste* my time on things like this.
(6) You shouldn't *spend* any time on that.

The ontology of time in any system based on these logics clearly must be augmented if they are to be able to deal with how we talk about time. In the case of English, the metaphor structuring time as a resource is necessary.

The second situation comes about when the knowledge–base designer is forced to rely heavily on linguistic evidence from a single language for the representation of a particular common–sense domain. In this case, the resulting representation may simply be an implicit representation of a conventional metaphor for the domain for a given language. (Indeed for many applications this may be sufficient). This problem is most apparent when one considers the problem of translation.

The MetaBank project provides a partial solution to both of these problems. In the first situation, MetaBank provides the connection between the representations of domains and how the domains are expressed in language by augmenting domain representations with representations of conventional metaphors. In the second situation, MetaBank provides a motivated role for linguistic data that does not force an entirely linguistic representation.

8 Conclusions

The frequent and conventional use of non–literal language remains a major problem for natural language processing systems. Knowledge–intensive methods based on explicitly represented non–literal language conventions have recently made progress towards a solution. The MetaBank project has taken the first steps towards putting these approaches on a firm theoretical and practical foundation by providing a robust knowledge–base of metaphorical conventions.

This article has introduced our three–part approach to the construction of MetaBank: the collection of on–line textual resources and databases of linguistic generalizations, the development of a methodology for analyzing these resources, and the construction of a knowledge–base based on the preceding analyses. The initial results from our corpus analysis indicate that metaphor is indeed a frequent phenomenon and that a small number of core metaphors underlie the large number of frequently occurring metaphors.

References

Brachman, R. J. and Schmolze, J. (1985). An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:346–370.

Brill, D. (1992). *LOOM Reference Manual Version 1.4*. USC Information Sciences Institute.

Fass, D. (1988). *Collative Semantics: A Semantics for Natural Language*. PhD thesis, New Mexico State University, Las Cruces, New Mexico. CRL Report No. MCCS-88-118.

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

Genesereth, M. R. and Fikes, R. E. (1992). *Knowledge Interchange Format Version 3.0 Reference Manual*. Computer Science Department, Stanford University.

Gruber, T. R. (1992). *Ontolingua: A mechanism to Support Portable Ontologies*. Knowledge Systems Laboratory, Stanford University. Version 3.0.

Hammond, K. (1986). *Case Based Planning: An Integrated Theory of Planning, Learning and Memory*. PhD thesis, Yale University. Department of Computer Science Report No. 488.

Helman, D. H., editor (1988). *Analogical Reasoning*. Kluwer Academic Publishers, London.

Hinkelman, E. (1989). *Linguistic and Pragmatic Constraints on Utterance Interpretation*. PhD thesis, University of Rochester, Computer Science Department, Rochester, NY. Report No. 288.

Hinkelman, E. (1990). Surface signals of illocutionary intentions. In *Proceedings of the 5th Rocky Mountain Conference On Artificial Intelligence*.

Hinkelman, E. and Allen, J. (1989). Two constraints on speech act ambiguity. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*.

Hobbs, J. (1979). Metaphor, metaphor schemata, and selective inferencing. Technical Report Technical Note 204, SRI, San Mateo, CA.

Hobbs, J., Croft, W., Davies, T., Edwards, D., and Laws, K. (1987). Commonsense metaphysics and lexical semantics. *Computational Linguistics*, 13(3).

Jackendoff, R. (1983). *Semantics and cognition*. MIT Press, Cambridge, MA.

Jacobs, P. S. (1985). *A Knowledge-Based Approach to Language Production*. PhD thesis, University of California, Berkeley, Computer Science Department, Berkeley, CA. Report No. UCB/CSD 86/254.

Kolodner, J. and Simpson, R. (1989). The mediator: Analysis of an early case–based problem solver. *Cognitive Science*, 13(4):507–549.

Lakoff, G. (1987). *Women, Fire and Dangerous Things.* University of Chicago Press, Chicago, Illinois.

Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By.* University of Chicago Press, Chicago, Illinois.

Lenat, D. B. and Guha, R. (1990). *Building Large Knowledge Bases.* Addison-Wesley.

Martin, J. H. (1990). *A Compututational Model of Metaphor Interpretation.* Academic Press, Cambridge, MA.

Martin, J. H. (1992). Computer understanding of conventional metaphoric language. *Cognitive Science*, 16(2):233–270.

McRoy, S. W. (1992). Using multiple knowledge sources for word sense discrimination. *Computational Linguistics*, 18(1):1–30.

Norvig, P. (1987). *A Unified Theory of Inference for Text Understanding.* PhD thesis, University of California, Berkeley, Computer Science Department, Berkeley, CA. Report No. UCB/CSD 87-339.

Pollio, H. R., Smith, M. K., and Pollio, M. R. (1990). Figurative language and cognitive psychology. *Language and Cognitive Processes*, 5(2):141–167.

Prieditis, A., editor (1987). *Analogica: Proceedings of the First Workshop on Analogical Reasoning*, Los Altos, CA. Morgan Kaufmann.

Reddy, M. J. (1979). The conduit metaphor. In Ortony, A., editor, *Metaphor and Thought*, pages 284–324. Cambridge University Press.

Talmy, L. (1975). Semantics and syntax of motion. In *Syntax and Semantics*, volume 4. Academic Press, New York.

Talmy, L. (1988). Force dynamics in language and cognition. *Cognitive Science*, 12(1):49–100.

Wilensky, R. (1986). Some problems and proposals for knowledge representation. Technical Report UCB/CSD 86/294, University of California, Berkeley, Computer Science Division.

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

Zernik, U. (1987). *Strategies in Language Acquisition: Learning Phrases from Examples in Context.* PhD thesis, University of California, Los Angeles, Computer Science Department, Los Angeles, CA.

Zernik, U. (1991). Tagging word senses in a corpus. In Zernik, U., editor, *Lexical Acquisition*, pages 91–112. Lawrence Erlbaum Associates.

Zipf, G. (1949). *Human Behavior and the Principle of Least Effort.* Addison–Wesley, Cambridge, MA.