

An Assessment of Cyc for Natural Language Processing

Kavi Mahesh, Sergei Nirenburg, Jim Cowie, and David Farwell
Computing Research Laboratory
New Mexico State University

Abstract

This is the final report on the assessment of Cyc for natural language processing applications. The work reported here was carried out by the authors at CRL, NMSU under collaboration with both the Department of Defense and Cycorp, Inc. The primary motivation of this relatively small-scale exercise was to arrive at an independent assessment of the utility of Cyc's knowledge and inference capabilities for solving difficult problems in NLP and machine translation. Word sense disambiguation and coreference resolution were chosen as the two problems for this study. We conclude from this exercise that Cyc in fact has a large amount of knowledge that is potentially useful for solving these problems in NLP. However, the knowledge in Cyc is not directly applicable to the problems either in an exclusively Cyc-based solution or one where Cyc is used to improve the performance of other methods. In this report, we have attempted to identify the primary reasons why Cyc cannot readily solve NLP problems, to illustrate our findings with many real-world examples, and to suggest changes or enhancements to Cyc that might make its knowledge more readily applicable to NLP problems.

1. Motivation

Cyc is one of the largest representations of general world knowledge ever constructed (Lenat and Guha, 1990; Lenat, et al, 1990; Whitten, et al, Version 2.0). Natural language processing (NLP) is an important problem in information processing that requires large amounts of broad-coverage world knowledge. This project brought the two together to assess the utility of Cyc for solving central problems in NLP.

This report describes our analyses and experiments for assessing the utility of the knowledge present in the Cyc knowledge base for solving well-known problems in natural language processing and machine translation (MT, see, e.g., Nirenburg, et al, 1992) such as *word sense disambiguation* and *coreference resolution*. These two problems were chosen for evaluating Cyc because of their requirements for large amounts

of general world knowledge of the kind present in Cyc and the need for drawing chains of inferences in order to select the right word sense or coreferent.

2. Analysis and Experimentation: An Overview

The methodology followed in this assessment involved both critical analysis and small-scale experimentation. The problems of *word sense disambiguation* and *coreference resolution* were analyzed using examples from a corpus to determine the requirements they place on a knowledge base and inference mechanism such as Cyc. Such requirements included specifications of types of knowledge, their coverage, accessibility and usability of the knowledge, and different kinds of inferences to be made from the knowledge. With these requirements in mind, detailed explorations of Cyc were conducted manually to assess the utility of Cyc in solving many examples of both the disambiguation and coreference problems found in real-life texts from the corpus.

Apart from the above analyses, several small-scale experiments were conducted to acquire a small lexicon using Cyc and to solve different kinds of disambiguation and coreference problems using the knowledge in Cyc. Below, we report on our findings from these empirical evaluations and make suggestions for further work in applying Cyc for solving practical problems in NLP and MT.

The above evaluations address Cyc's ability to meet some of the requirements of NLP. This report is not concerned with the implications of using Cyc on other components of a practical NLP or MT system.

3. The Problems: Resolving Ambiguity and Coreference

It must be noted that solving either of the following problems requires one to formulate appropriate queries from the input texts. In this study, we assume for the most part that this problem of "going from the texts to their meaning representations" can be solved, somehow. In the case of word sense disambiguation, we have a working semantic analyzer in the Mikrokosmos project (Beale, et al, 1995) and have used it as a rough guide in determining the nature of the queries to be answered by Cyc. Queries in coreference problems are more complex, involve lengthy chains of inferences, and the generation of queries from input texts has been done mostly by hand.

3.1 Resolving Word Sense Ambiguity

Word sense disambiguation is a central problem in NLP and MT. In translation, finding the right words in the target language for words in the source text depends critically on being able to select the right sense of a source word. Knowledge-based methods offer a

solution to word sense disambiguation that has the potential of achieving higher accuracy than what is possible from other methods, such as corpus-based statistical methods.

The basic problem of word sense disambiguation is to select the combination of word senses for all the words in a sentence (or an entire text) that best fits the overall meaning and context of the sentence. The best combination may be selected by determining how well a given sense of a word combines with senses of other words in the sentence to form a coherent meaning for the entire sentence (or text). Such **measurements of semantic affinity or conceptual distance** between word senses requires a large body of common sense knowledge that is designed to enable efficient inferences for making the measurements.

Word sense ambiguities can be classified into three types:

- One sense fits the context; rest are anomalous;
- Two or more senses are acceptable; must choose from them somehow;
- All senses are anomalous; must choose from them nevertheless.

An ideal case of word sense ambiguity is one where all but one of the senses of a word are *anomalous* when combined with the possible senses of other words in the sentence. That is, all but one sense violate some known constraint on the relationships among the meanings of words in the sentence. For example, consider the Spanish sentence¹

`"Fuentes financieras consultadas cifraron..."` (Financial sources that were consulted estimated....)

The word "fuente" has three senses: a *source*, a *fountain*, or a *plate*. In this case, we expect our general world knowledge to tell us that the *fountain* and *plate* senses are anomalous since they cannot be *consulted*. A knowledge base such as Cyc must be able to provide such knowledge in order to be useful for word sense disambiguation. That is, Cyc must have a selectional constraint on who/what can be *consulted* and the constraint must include *sources* but not *fountains* or *plates*, for the above example to be an ideal case.

Often more than one sense of word meets known selectional constraints. For example, consider the well-known sentence

`"The box is in the pen."`

¹. Many of our Spanish examples came from news articles on company mergers and acquisitions. We are currently working on these texts in the Mikrokosmos project. Many of our English examples of word sense ambiguity are well known in NLP literature. English texts used for the coreference experiments came from the MUC-6 evaluation (MUC6, 1995).

The “pen” could be a *writing pen*, an *animal pen*, or a *playpen*. Our knowledge of semantics may only provide constraints that say that the “pen” must be a physical object that can function as a container. All three senses meet this constraint. Selecting the right sense of “pen” requires additional world knowledge in the form of default sizes of *boxes*, *animal pens*, and *playpens*. Such knowledge may enable us to infer that a *writing pen* is perhaps too small to contain a *box* in it and hence the “pen” must be one of the other two possible interpretations. In fact, in the absence of additional context, it may not be possible to eliminate either of the remaining two senses.

The most interesting cases of word sense disambiguation are those where all senses are rejected by known constraints but we still need to choose the best sense. For example, given a sentence such as (Onyshkevych, 1995)

“John drove his V8 down the main street.”

we need to find out if the *V8 engine* sense of “V8” fits the *drive vehicle* sense of “drive” better than the *V8 vegetable juice* sense fits *drive vehicle*. Of course, our knowledge of selectional constraints on *drive vehicle* will tell us that only *vehicles* can be driven, thereby rejecting both available senses of “V8”. We must employ more detailed world knowledge which might tell us that “*V8 engine* is a type of *engine*; *engines* are parts of *automobiles*; an *automobile* is a type of *vehicle* and hence is something which a *person* can *drive*.” And, the knowledge base might tell us that any relationship between *V8 vegetable juice* and *drive vehicle* is much longer (in some measure of distance). If such knowledge is available, then we can resolve the ambiguity and correctly select the *V8 engine* sense. Note that this involves not only having all the of necessary knowledge in the knowledge base but also being able to access the right pieces to identify relationships between concepts through one or more previously unknown, intermediate concepts (e.g., part-of and automobile). Constraining this search so that only relationships meaningful and relevant in a given context are selected is a separate problem that we are addressing currently in the Mikrokosmos project.

It may also be noted that the above sentence is also an example of metonymy (e.g., Fass, 1988) where the engine was used in place of the vehicle that has the engine as a part. Non-literal usage such as metonymy and metaphor are extremely common in real-world texts and their resolution is more or less inseparable from word sense disambiguation. We are also investigating metonymy, metaphor, and other non-literal expressions under a separate IDEAS project.

3.2 Resolving Coreference

Coreference resolution, or more properly reference resolution, is the process of establishing a link between an expression which is being used to refer to something and whatever it is it is being used to refer to. For a text (segment) such as:

As senior director of UAL, and a member of the executive committee of its board, I am appalled at the inaccuracies and anti-management bias in the Journal's April 17 article about Richard Ferris, UAL's chief executive officer.

The UAL board is four-square behind Mr. Ferris, his management team and his long-range strategy of making United a more competitive airline by combining it with the premier hotel company and car rental company.

it is necessary to establish that "UAL" in "senior director of UAL", "it" in "its board", "UAL" in "UAL's chief executive officer", "UAL" in "the UAL board" are all being used to refer to the same company, that "Richard Ferris", "UAL's chief executive officer", "Mr. Ferris", "his" in "his management team" and "his" in "his long-range strategy" are all being used to refer to the same person, that "United" and "it" in "combining it with..." are both being used to refer to the same company and that "its board" and "the UAL board" are both being used to refer to the same group. At the same time, it is necessary to recognize that, say, "the premier hotel company" and "[the premier] car rental company" are not being used to refer to either UAL or United or that "UAL's chief executive officer" is not being used to refer to the senior director of UAL and so on. The task is elementary for virtually any NLP application but certainly for such applications as machine translation, information extraction, summarization or interactive interfaces for database query systems, expert systems, tutoring systems and so on.

Below, we outline a system which is under development at the CRL for establishing links between coreferring expression in a text and identify a number of problems that we feel Cyc might be applied to in order to improve the performance. This system was developed to handle the limited forms of coreference specified for the Sixth Message Understanding Conference (MUC6, 1995). We first briefly describe the current system and its performance with respect to a specific document. We then describe planned modifications to the current system and indicate the expected improvements in performance that should follow. Finally, we present certain types of problems which we do not expect the revisions will take care but rather which appear to require inferencing in part on the basis of common knowledge of the sort Cyc should be able to provide.

3.2.1 Current Core Algorithm

The current algorithm for establishing links between expressions that are used to refer to the same object is divided into two phases. The first focuses on building up a pool of potential referents on the basis of expressions that are highly likely to be used as referring expressions. These include the names of companies, the names of countries, the names of provinces, the names of cities, the names of people, years, the names of government organizations and noun phrases which are either not definite singular or are definite singular but do not match any of the previously established referent names. Thus, as each potential referring expression (noun phrase or proper noun) is inspected, the procedure will trigger the creation of a new referent if the expression is of any of the types listed

above and it does not match the name of an existing referent. The second focuses on identifying expressions that are highly likely to be used to corefer to an existing referent and on linking those expressions to their referents. These include the definite singular NPs, pronouns, certain 0-determiner NPs related to job positions, and appositives.

The linking of proper nouns is done by accessing the most recently mentioned referent of the same type (person, company, etc.) and matching the associated referring expressions using either exact match, case insensitive match, substring containment or abbreviated form match. For definite singular NPs, the head of the NP is first matched against the semantic type (person, company, etc.) of a potential referent, or, failing that, it is matched against a list of typical heads of expressions that are used to corefer to referents of the given semantic type (e.g., "cartel"-headed NPs are often used to refer to companies). For the remaining NPs, the head is (exact) matched against the heads of the expressions used to refer to the established referents. For the remaining referring expressions, that is, pronouns and job positions, the most recently mentioned referent of a pre-determined type (person for "he", "she", "you", and so on or company or city for "it", etc.) is selected.

3.2.2 Current Performance

For the specific text that we inspected, there were 89 links, 22 chains (or referents) and 111 expressions related according to a trial MUC evaluation key. The CRL algorithm asserted 61 links of which 35 links were correct (35/89, 0.39 recall; 35/61, 0.57 precision) and 26 links were incorrect. The algorithm failed to assert 54 links.

3.2.3 Proposed extensions to the core algorithm

The above algorithm can be extended by first determining the syntactic category of the referring expression (i.e., proper noun, pronoun, or common noun) and then follow the procedure for that particular category.

For proper nouns, the basic resolution procedure should be to apply a string match against each of the proper nouns previously used (i.e. those that have already been used to established or refer to a referent in the domain of discourse). If there is no match, assume that it is not being used to refer to anything already in the domain and, therefore, add a new referent. This is essentially what the existing algorithm does in any case except that the semantic type restriction has been removed so that a given expression will be checked against all the proper noun expressions used thus far.

For pronouns, the basic resolution procedure should be to check each referent for "morphological" compatibility in reverse order of mention (recency). This represents a major generalization of the current algorithm which checks only those referents of a predetermined semantic type.

Thus the procedure is to take the sparse constraints provided by the form of the pronoun (e.g. "it" is used to refer to something that is singular or mass, non-human and neither speaker/author nor addressee of the text) and then check each referent, starting with the most recently mentioned, to see if it is compatible with those the constraints. If it is, assume that the pronoun is being used to corefer to that referent. If it is not compatible, check the next most recently mentioned referent.

Exceptions to this treatment include first and second person pronouns, which by default refer to the speaker (1st singular), the speaker's group (1st plural) and the addressee or addressee's group (2nd), relative pronouns which refer to the head of the relative construction. The former still involve establishing who the various discourse participants are, a process similar to that for 3rd person pronouns.

The major modification to the existing algorithm, however, concerns the treatment of common nouns where the focus of the procedure is shifted from semantic checks to string matching of referring expressions. Under the revised approach, the common noun (CN) resolution process begins by identifying whether the expression in question is an appositive. If it is, assume the expression is being used to corefer to whatever the phrase to which it is in apposition with refers to. If not, attempt to match the head of the expression with the heads of each of the CNs that have been used to refer (or corefer) thus far. If there is a match, assume the expression under consideration is coreferring. If not, check to see if the expression has a determiner with deictic force (demonstrative or definite quantifier such as "either", "both", "each", etc.). If it does, follow the basic pronoun resolution strategy of checking the referents, in reverse order of mention, for semantic compatibility. If not, attempt to match the complement of the expression under consideration with the complements of the CNs that have been used to refer thus far. If there is a match, then attempt to infer a semantic connection between the referents of the heads of the two expressions. The inferencing will be knowledge based and so Cyc is a potential knowledge source. If there is a connection, assume the expression is coreferring. If not, match the head of the expression under consideration to the PNs that have been used to refer thus far. If there is a match, attempt to infer a semantic connection between that referent and the potential referent of the expression under consideration. The inferencing in this case will also be knowledge based and so Cyc is again a potential knowledge source. If the connection can be established, assume the expression is coreferring. If not, assume the expression refers to something new and add that referent to the domain of discourse.

This basic common noun resolution procedure is constrained by the following heuristics. If the expression being processed is indefinite, assume it is non-coreferring. If the expression under consideration does not agree in grammatical number with the potential coreferring expression, assume it is non-coreferring. If the expression under consideration is used as part of a "fixed" expression, assume it is non-coreferring. If the expression under consideration is of the form "the X" and used referentially (to refer to a particular individual) and the potential coreferring expression is of the form "an X" and used attributively (to refer to a generic concept which is asserted to be a property of some sort), assume the former is non-coreferring. If the expression under consideration is of the

form "X" and the potential coreferring expression is of the form "the X", assume the former is non-coreferring.

3.2.4 Potential Improvements

Of the 54 unasserted links, 40 can be dealt with by the improvements to the core algorithm outlined above resulting in improved recall on this particular text (75/89, 0.84 recall). But the other 14 links appear to require inferencing supported by a knowledge base such as Cyc. Of the 26 falsely asserted links, 13 can be dealt with by the improvements to the core algorithm suggested above resulting in improved precision on this text (75/88, 0.85 precision). The 13 other cases, however, appear to require inferencing with support from Cyc.

4. Requirements

This section outlines some of the general requirements for a knowledge base to be used for NLP followed by specific requirements of word sense disambiguation and coreference resolution.

4.1 General NLP Requirements

4.1.1 Measuring Semantic Coverage for NLP²

How do we measure the coverage of an NLP system or a knowledge base being used for NLP? Current development in NLP is driven to a large extent by direct measures of knowledge-base size and coverage of individual phenomena relative to a corpus. These measures do not adequately motivate progress in computational semantics of natural languages. We propose **depth** and **breadth** as important measures of coverage in addition to **size** (Nirenburg, Mahesh and Beale, 1996). The amount of information (i.e., depth) and the types of information (i.e., breadth) contained in each element of a knowledge base are as important as the total number of elements (i.e., size) present. This claim is based on the following *scalability* issues that are an essential part of developing large-scale, general-purpose, NLP systems:

- domain independence: scalability to new domains; general-purpose NLP
- language independence: scalability across languages
- phenomenon coverage: scalability to new phenomena; going beyond core semantic analysis; ease of integrating component processes and resources.

². This sub-section is condensed from the article by Nirenburg, Mahesh, and Beale (1996) which appeared in COLING-96.

- application-independence: scalability to new applications; toolkit of NLP techniques applicable to any task.

We believe that coverage in terms of the depth and breadth of the knowledge given to an NLP system is mandatory for attaining the above goals in the long run. Such coverage is best estimated not in terms of raw sizes of lexicons or world models but rather through the availability in them of information necessary for the treatment of a variety of phenomena in natural language (i.e., breadth and depth)---issues related to semantic dependency building, word sense disambiguation, semantic constraint tracking and relaxation (for the cases of unexpected input, including non-literal language as well as treatment of unknown lexis), reference, pragmatic impact and discourse structure. The resolution of these issues is at the core of post-syntactic text processing.

There exist other, broader desiderata which are applicable to any information processing system. They include concerns about system robustness, correctness, and efficiency which are orthogonal to the above issues. Equally important but more broadly applicable are considerations of economy and ease of acquisition of knowledge sources --- for example, reducing the size of knowledge bases and sharing knowledge across applications.

How do we measure the coverage of a knowledge base for NLP? A useful measure of semantic coverage must involve measurement along each of the three dimensions: depth, breadth, and size. Our experience over the years has led us to the following sets of criteria for measuring semantic coverage. However, we understand that the following are not complete or unique; they are representative of the types of issues that are relevant to measuring semantic coverage:

- the number of properties or links defined for an individual concept
- number of types of non-taxonomic relationships among concepts
- average number of links per concept: “connectivity”
- types of knowledge included: defaults, selectional constraints, complex events (e.g., scripts), etc.
- ratio of number of entries in a lexicon to number of concepts in the ontology
- and, finally, total number of concepts in the ontology.

4.1.2 Lexical Requirements

Before Cyc can be used for solving semantic problems in NLP, there must be a lexicon that maps words in the desired source language to concepts in Cyc. Cyc has a partial lexicon for English which is currently under development. For experimental purposes, a very small Spanish lexicon was constructed by substituting Cyc concept names in the existing semantic mappings to the Mikrokosmos ontology in a small part of the

Mikrokosmos Spanish lexicon. Acquiring a lexicon by mapping words to Cyc requires the following:

- there must be a *sufficient number of concepts* covering a broad range of domains so that meanings of all words can be expressed without excessive or unwieldy decomposition of meanings;
- *concepts must be easy to find* given only a vague description of the desired meaning (i.e., a *gloss*); lexicon acquirers cannot be expected to be completely familiar with the contents of a large knowledge base such as Cyc;
- there must be a *sufficient amount of knowledge about a concept* and about its relationships with other concepts (i.e., sufficient depth and breadth); mere labels or a mere taxonomy is not sufficient for representing meanings of all words;
- *knowledge about a concept must be easily accessible*; it must be easy to *see* everything known about a particular concept (including any knowledge present implicitly through inheritance or other forms of inference); only then can a lexicographer decide if it is the right concept for representing the meaning of a word and, if so, what additional modifications or constraints are needed to adequately capture the meaning of the word in that language; and finally,
- the *knowledge representation must be sufficiently expressive* to allow meanings of words to enhance or constrain the meanings of other words in various ways.

4.2 Requirements for Word Sense Disambiguation

The basic mechanism in knowledge-based word sense disambiguation is applying selectional constraints on relationships between concepts stored in the knowledge base to eliminate as many senses as possible. This mechanism places the following requirements on the knowledge base (Mahesh and Nirenburg, 1996):

- concepts must be well-connected to other concepts in the ontology; there must be a variety of relations for linking concepts to one another;
- every relation between a pair of concepts must have a well-defined constraint (i.e., a domain and a range of concepts that are acceptable for the relation);
- all concepts must be organized in a taxonomy so that constraints can be stated concisely (i.e., using intermediate nodes in the taxonomy to refer to entire sets of concepts, rather than having to enumerate all permissible concepts);
- any attributes of concepts to be filled by numbers or literal constants must also have constraints specified (e.g., a permissible range of numbers or an enumeration of literal constants); and
- the knowledge base must support *taxonomic queries* for checking constraints (i.e., queries of the form “Is Concept1 a subtype of Concept2?”).

For example, the concept `ConsultingAnExpert` must specify who can consult, who can be consulted, for what purpose, and so on. That is, the concept must have relations such as

agent (or performedBy) that constrains “the who” to some subset of all concepts in the knowledge base. Such information will help us rule out the *fountain* and *plate* senses of the “fuente” in the sentence “Fuentes financieras consultadas cifraron...” (see Section 3.1).

As noted earlier (in Section 3.1), selectional constraints do not always retain exactly one sense for each word. When zero or two or more senses are accepted by known selectional constraints, we must be able to determine which of the available senses meets the constraints better than others. In order to support this, the knowledge base must support

- *quantitative taxonomic queries*, that is, queries of the form “How well is Concept1 a subtype of Concept2?”.

In the case of zero choices, the above query must be answered even though Concept1 *is not* a subtype of Concept2. That is, the query should be read in this case as “How well does Concept1 meet the constraint that it must be a Concept2?” Since there is no sequence of taxonomic links that relates the two concepts in such a situation, the knowledge base must be able to determine some other sequence of links and provide a measure of the *distance between the two concepts* (which is also called the *semantic affinity* between the concepts).

As seen earlier in the “V8” example (in Section 3.1), both senses of “V8” are rejected by a reasonable selectional constraint on “the who” of *drive vehicle*. In order to disambiguate “V8”, Cyc must be able to search for direct or indirect relationships between *V8 engine* and *drive vehicle* and compare the distance (or cost) of this path with another path between *V8 vegetable juice* and *drive vehicle*. It is important to note that the types of relationships along these paths are unknown at this point; Cyc must be able to search for the relationships that result in the shortest path between the concepts.

Such situations occur regularly in any non-trivial NLP system because of imperfect constraints and routine problems in texts such as vagueness and non-literality (metonymy and metaphor). It is only in a small number of sentences that all known constraints are met literally. Experience with real-life texts (such as the news articles in our corpus) shows that:

- seemingly valid constraints are violated routinely by intended interpretations of real-world sentences;
- it is almost impossible to write precise constraints (or otherwise structure the knowledge base) to fit just the intended interpretation of previously unseen texts, while still retaining the power to disambiguate (i.e., to not accept too many interpretations). Many real-world sentences are in fact “not so literal” with respect to a knowledge base although they may appear to be completely literal to people.

We would like the knowledge base to tell us something useful when constraints *are* violated. If we have n interpretations all of which violate one or more constraints, we

would still like to be able to prefer one or two of the n over others based on available world knowledge.

Queries of the form “Are Concept1 and Concept2 related (through some unknown relation)? If so, how closely?” are also essential for solving other problems such as interpreting compound nouns. It is unreasonable to expect the knowledge base to provide a direct relation every pair of concepts that are ever related to one another in a text. For example, in “The box is in the pen”, there may not be any direct relation between *BoxTheContainer* and any of the senses of “pen”. If the knowledge base had a *fitsIn* relation that linked the two concepts with an associated constraint that ruled out *WritingPen* from the range of *fitsIn* for *BoxTheContainer*, while allowing *AnimalPen* and *play pen*, we could disambiguate “pen” in no time. However, one must not expect to find such direct hits for every input nor try to acquire such specific constraints. Instead,

- the knowledge base must contain default values (or ranges of values) for a variety of attributes of concepts;

For example, knowing the default sizes of “boxes” and “pens” and having a constraint on *fitsIn* that specifies the relative sizes of objects that can fit into other objects results in a more feasible solution to the disambiguation problem.

In summary, apart from *taxonomic* and *quantitative taxonomic queries*, Cyc must be able to answer the following types of queries to support word sense disambiguation:

- *Direct-link-unknown-relation queries: Is there a relation (or “link”) between Concept1 to Concept2?*
- *Quantitative direct-link unknown-relation queries: What is the strongest relation between Concept1 and Concept2? If Concept1 is the chosen sense of a word and Concept2 and Concept3 are possible candidates for another word, queries of this kind help select one of Concept2 and Concept3, depending on which of them has a stronger relation to the already chosen Concept1.*
- *Arbitrary path queries: Is there any relation (direct or transitive, i.e., a path) between Concept1 and Concept2 (through one or more intermediate relations that are not known beforehand)? These queries are essential for disambiguation when there is no choice that meets the known constraints literally.*
- *Quantitative arbitrary-path queries: What is the strongest relation (or least-cost path) between Concept1 and Concept2? These quantitative measures help pick the closest sense of a word when there is none that meets the known constraints literally.*
- *Constrained pattern path queries: Is there any relation (or what is the strongest relation) between Concept1 and Concept2 that follows the given pattern of link types (i.e., a sequence of relation types)? These queries help constrain the search for paths and also focus on looking for particular phenomena in natural language semantics such as metonymies and idiosyncratic constraint relaxations in particular languages.*

It may be noted that other problems in NLP and MT, such as *coreference resolution*, require many of the same types of queries.

4.3 Requirements for Coreference Resolution

Potential Cyc applications:

As noted above (Section 3.2.), there are potentially 27 cases (of 80 current problems) that Cyc can help in resolving. Note that it is not unusual in performing needed inferencing that a general knowledge of the objects and processes referred to in a text, as might be found in Cyc, must be used in conjunction with facts that are explicitly being reported in the text. We present the different cases along with our findings in trying to make the relevant inferences using Cyc, in Section 5.3 below.

It must be noted here that the requirements of the coreference problem in terms of the types of knowledge and inference needed are remarkably similar to those of word sense disambiguation. Noun phrases in texts that need to be corefered often map to different concepts (where one concept bears a taxonomic or other metonymical relationship to the other). For example, if one of the coreferents is the name of a company, the other may be a definite expression such as “the company,” “the firm,” etc. or an alias or acronym of the company, and so on. In such cases, the relationship (if any) between the two concepts will not be known beforehand. The coreference resolver needs to determine if there is a direct or indirect relationship between the two concepts and how close or strong the relationship is. Doing this requires Cyc to answer unknown-predicate, arbitrary-path, quantitative queries.

5. Findings and Results

The first part of this section reports on our findings on the semantic coverage of Cyc for NLP in terms of measures of size, depth, and breadth of knowledge contained in it. The subsections that follow report on the disambiguation and coreference experiments with ample examples to illustrate our findings.

5.1 Semantic Coverage of Cyc for NLP

Cyc has the following types of knowledge:

- Taxonomic knowledge: *genls* and *instanceOf* predicates.
- Attributes of individual concepts
- Relations between concepts and constraints on them (represented in the form of axioms)

- Instances of concepts
- Lexical knowledge

5.1.1 Coverage of Cyc Knowledge³

We found 21078 terms in Cyc, not all of which may be considered concepts.⁴ Our objective here was to measure the ontological coverage of Cyc in terms of the number of terms that can be considered separate concepts and compared with similar numbers from other ontologies. For example, we did not consider instances of concepts as separate concepts.

There are many instances of concepts (although Cyc does not distinguish between concepts and instances in measuring the sizes of its knowledge base). For example, Alabama-State, CityOfMadrasIndia, CorazonCAquino, SayidMohammadNajibullah, SoutheasternUniversityOfTheHealthSciencesCollegeOfOsteopathicMedicine, etc. cannot be counted as concepts. There are also concepts that could be easily obtained by composing other concepts. For example, AvailableForDating, StandingWithLegsCrossed, SvalbardIslandsPerson, SwimmingTheLengthOfOlympicPool, etc. can all be obtained compositionally from other concepts. Hence, they were not counted as separate concepts. All of these observations lead us to believe that there are about 10,000 - 12,000 concepts in Cyc at present.

These concepts indeed cover a very wide range of domains. Yet, there are inevitably gaps in the conceptual coverage. For example, no concepts could be found for “playpen”, “sacrifice”, “devotion”, “adopt”, “beg”, “tea bag”, etc. Although there are many instances of concepts in Cyc, the gaps in coverage seem to be much larger in instances. In any category, such as languages, cities, religions, companies, etc., many missing instances can be listed. For example, AmericanAirlines is the only airline listed and AlamoAutoRentalCompany is the only car rental company listed. Although it was perhaps never the intention of Cyc to cover a large set of instances within the Cyc knowledge base, the coverage is rather non-uniform among the instances that are included in Cyc.

Each concept, on an average, has a substantial amount of knowledge. However, the breadth of such knowledge in terms of the types of knowledge included appears to be more or less random. For example, consider the concept AirplanePilot. There are seven

³. All numbers and examples presented in this report refer to the latest version of Cyc available to us at the time of reporting. This is the version installed in CRL on May 21, 1996. Most of the examples used were encountered when trying to analyze cases of word sense ambiguity and coreference in our texts.

⁴. The 21078 terms were extracted from Cyc by looking for all instances of the predicates *comment*, *denotation*, *genls*, and *instanceOf*. Words in Cyc’s lexicon (i.e., *-TheWord) were not included in this count.

assertions in which this concept participates (apart from taxonomic ones). Of these, four are (in a simplified format):

```
AirplanePilot => (income (DollarsPerYear 70000
100000))

AirplanePilot => (socialClass UpperMiddleClass)

AirplanePilot => (imageDesired SexyPersonality)

AirplanePilot => (educationLevel UndergraduateLevel)
```

It is not clear why only these facts were included. Why not facts about typical age and good vision? The above facts do not constitute what a typical encyclopedia contains about AirplanePilots. Moreover, the above type of information may not be very useful for NLP purposes. How often do we need to infer that “he/she” must be an AirplanePilot because “he/she” makes 70000 or “he/she” has SexyPersonality? If the above facts are included, dozens of other similar facts must also be present to obtain uniform coverage in terms of breadth and depth of knowledge for each concept.

Another example of incomplete coverage (in the form of non-uniform breadth) can be found in our earlier example “The box is in the pen” (Section 3.1). The word “pen” is mapped to WritingPen and PigPen in Cyc’s lexicon. There is no mapping to “playpen” nor is there a concept for it.⁵ WritingPen has knowledge of its typical size:

```
(=> (iO $U WritingPen)

(fitsIn $U (Cylinder-Function (Centimeter 20)

(Centimeter 2))))
```

No such knowledge is present in PigPen. Thus, the above piece of knowledge about WritingPens while relevant to our current example is not useful for disambiguation due to non-uniform breadth of coverage.

Cyc lexicon has:

⁵. It may also be noted that although a more general concept called AnimalPen is available, “pen” is mapped to PigPen. Incidentally, PigPen is *not* a specialization of AnimalPen.

- 12860 entries: an entry here is an instance of the “denotation” predicate; this, we believe, is a rough mapping from a word to a concept, not the “lexical semantics” of the word.
- The 12860 entries cover 9131 word forms
- Of the 12860 entries, about 3800 have no mappings to Cyc concepts currently (i.e., they map to StandInDenotation, to be filled in later when the necessary concepts are acquired).
- Of these, 693 are proper nouns (“proper adjectives” are also listed as proper nouns).
- A test using the list of English words in Unix’ /usr/dict/words showed that only about 4500 out of the 25000 words had entries in Cyc’s lexicon.
- As for detailed semantic mappings (other than just “denotations”), we found
 - 458 mappings for verbs (the verbSemTrans predicate) covering 382 verbs
 - 212 mappings for nouns (the nounSemTrans predicate) covering 197 nouns
 - 318 mappings for adjectives (the adjSemTrans predicate) covering 281 adjectives
 - 0 mappings for adverbs (the adverbSemTrans predicate)

There are small numbers of other mappings:

- 4 uses of the agentiveNounSemTrans predicate
- 74 uses of the possessiveSemTrans predicate
- 22 uses of the lightVerb-TransitiveSemTrans

The lexicon has moderate breadth; it includes a variety of predicates for representing morphological, syntactic, and semantic information. However, only a few entries have depth as can be seen from the above data. Most entries at present have direct mappings to concepts using the denotation predicate. The semantics of this predicate is not clear to us. For example, Traffic-TheWord is mapped to both Traffic and TrafficJam. The significance of the second mapping is not clear. For instance, why is it not mapped to TrafficLight or other senses of Traffic-TheWord such as in trade?

There are fine-grained sense distinctions in Cyc’s lexicon, yet the coverage of senses for each word is not “complete.” For example, Take-TheWord is mapped to TakingSomething in its denotation and it has other detailed mappings to TakingABath, TakingAShower, and TakingATransportationDevice (all using the lightVerb-TransitiveSemTrans predicate). While TakingABath and TakingAShower are distinct mappings, most other senses of Take-TheWord (e.g., taking a pill, taking a lesson, etc.) are not listed.⁶

⁶. See the article by Nirenburg, Raskin, and Onyshkevych (1995) for a discussion of how to determine how many senses should be there for a word in a computational lexicon.

There is no clear separation between knowledge of English words in the lexicon and knowledge of concepts in Cyc. Some rules that are better stated at the level of concepts are stated using individual English words. For example, although the following rule is probably intended to serve as a lexical rule that generates mappings from words to concepts, it is perhaps stated at the wrong level:

```
(=> (and (genls $DENOT Biting)

      (denotation $VERB Verb $SN $DENOT))

  (verbPrep-Transitive $VERB In-TheWord

    (and (iO :ACTION $DENOT)

          (toLocation :ACTION :OBLIQUE-OBJECT)

          (objectActedOn :ACTION :OBJECT)

          (anatomicalParts :OBJECT :OBLIQUE-OBJECT)

          (doneBy :ACTION :SUBJECT))))
```

i.e., for any Biting type verb followed by the English word “in” the toLocation of the action is an anatomicalPart of the objectActedOn.

Is this rule meant to serve as a phrasal entry for “bite x in y”? Currently Cyc has such rules for HittingAnObject, Kicking, and Piercing in addition to Biting. Are there going to be such entries for the collocations of In-TheWord with *every* other type of event? It appears that these rules should have been at the level of concepts such as Biting; the job of In-TheWord is merely to fill the toLocation predicate. That is, the rule should have been:

```
for any Biting type action the toLocation of the action is an
anatomicalPart of the objectActedOn.
```

This rule has nothing to do with the use of In-TheWord in English. If the intention was to capture the “collocation” (if any) between bite-type verbs and In-TheWord, a separate rule should have been stated at the level of English words:

```
for any Biting type verb followed by the English word “in” the
toLocation of the action is the :OBLIQUE-OBJECT.
```

This lexical rule has nothing to do with the conceptual relationships between anatomicalParts and Biting. Far fewer rules will be needed if conceptual and linguistic knowledge are kept separate this way and the coverage will also be more uniform. Moreover, rules such as the one above that make unnecessary use of particular words

(e.g., In-TheWord) make it very difficult to extend the use of such knowledge for processing languages other than English.

If the intention was indeed to capture a more specific rule:

```
for any Biting type verb followed by the English word "in" the
toLocation of the action is the :OBLIQUE-OBJECT (if and) only if the
:OBLIQUE-OBJECT is an anatomicalPart of the objectActedOn.
```

we believe that is an impossible goal to achieve. There are simply too many such “collocations” to acquire in the form of elaborate rules like the above before any degree of useful, uniform coverage can be attained for English or any other natural language.

5.1.2 Connectedness of Concepts

We are concerned that Cyc has constraints only on arguments of predicates.⁷ If you take a concept such as EatingEvent, there does not seem to be any constraint on its theme (i.e., objectActedOn). We were expecting to see a rule that constrains the objectActedOn to FoodAndDrink but didn’t find any. In fact, to deal with metonymy, we will need a second level of constraints that allows the above constraint to be relaxed to any container to permit such usage as “He drank the bottle.” Similarly, for the concept ConsultAnExpert, there are no constraints on who can consult, who can be consulted, or for what purpose. As such, there is no knowledge available to disambiguate “fuente” in our earlier example (from Section 3.1) “Fuentes financieras consultadas cifraron...”

In fact, very few concepts have relations to other concepts with constraints. One example where such a constraint is present is the concept Reading:

```
(=> (and (iO $X Reading) (objectInvolved $X $Y))
      (iO $Y TextualMaterial))
```

Unless such constraints are available regularly for most concepts, other knowledge about the concepts will not be of much use for word sense disambiguation. Of course, one could add such constraints to Cyc. We, however, are evaluating Cyc’s usefulness for solving MT problems based on the knowledge it has at present.

Apparently, many constraints that were present in older versions of Cyc were thrown out to improve the efficiency of inference processes (since they were of the form “For every EatingEvent there exists a Food that is the objectActedOn” instead of saying “If \$X is the

⁷. Less than 1500 of the 10,000 - 12,000 concepts in Cyc are predicates. Moreover, constraints on arguments of predicates are typically loose (see Section 5.1.3 below).

objectActedOn in an EatingEvent, then \$X is a Food”). We understand that efforts are underway currently to bring back such constraints and to make them accessible centrally from the concepts rather than being spread throughout the “sea of logical assertions.”

5.1.3 Usability of Cyc Knowledge

The key questions in usability of Cyc knowledge are (a) how easy it is to access efficiently the right information relevant to solving a particular problem in a particular context and (b) to what extent is the accessed knowledge directly usable for solving the given problem.

Even if Cyc can efficiently retrieve just the required knowledge, such knowledge may be untuned for the task on hand. For example, although there are selectional constraints in Cyc, it is not clear how effective those constraints are for disambiguating word senses in a real text. The constraints could be “too tight” for NLP purposes, that is, the knowledge in Cyc may exclude many of the intended interpretations of a text. Or, the constraints may be “too loose” and hence ineffective for disambiguation. Only an empirical study (perhaps on a scale much larger than what is possible in the current effort) can tell. For example, constraints specified on predicates themselves are in general too loose. For the predicate objectActedOn, we have:

```
argumentOneType : SomethingOccuring
```

```
argumentTwoType : SomethingExisting
```

Such loose constraints cannot disambiguate “fuente,” for example. For each event that has an objectActedOn, a much narrower range must be specified for permissible argumentTwoType. The current version of Cyc neither seems to have such constraints nor does it even say whether concepts such as EatingEvent or ConsultingAnExpert have an objectActedOn at all.

A basic problem of usability in Cyc (the current logic-based versions, not the frame-based versions of the eighties) is that **every piece of knowledge present in Cyc about a given concept is not accessible** at all.⁸ This is a problem for lexicon acquisition (as noted earlier in Section 4.1.2) and for formulating queries to make necessary inferences for

⁸. One can see all the rules in which the concept is present explicitly. It is also possible to go up the hierarchies and examine such rules for each of the ancestors of a concept. However, this amounts to leaving the burden of inheritance entirely to the user. It is humanly impossible for the user to manually examine all ancestors of a concept and manually compute all inherited and inferred properties of the concept in a large-scale knowledge base. The knowledge-base and its underlying representation must support such inheritance so that **everything known about a concept** can be seen immediately. Seeing and digesting such existing content is necessary for using the knowledge in lexical or other knowledge acquisition.

disambiguation, coreference, and other problems in natural language semantics. In fact, we wonder how knowledge acquisition in Cyc could be accomplished without such access. How could a new concept be added to the taxonomy or a relation added between a pair of concepts without seeing what knowledge is currently there (both explicitly and implicitly through inheritance and inference)?

Additional discussion of usability issues appears below in Section 6 in connection with issues in knowledge acquisition and the design of representations.

5.2 Disambiguation experiments

Most of our findings from the disambiguation experiments were reported throughout the previous section (Section 5.1). In addition to the above analyses, we built a small Spanish lexicon by mapping Spanish words in the “Roche text”⁹ to concepts in Cyc. The lexicon was built by modifying entries from the existing Mikrokosmos lexicon and covers the first three sentences in the text. Integrating the Mikrokosmos analyzer with Cyc to test disambiguation using Cyc’s knowledge is beyond the scope of the current effort. Instead, we looked at queries generated by Mikrokosmos in processing the sentences and explored Cyc manually to try to answer those queries. We found that:

- It is fairly easy to find concepts in Cyc to make direct mappings from words to concepts (as in the denotation predicates in Cyc’s lexicon).
- It is very difficult and expensive to find available knowledge about the concept in order to acquire detailed and precise mappings from words to Cyc expressions (Lenat and Guha, 1990).
- Cyc typically does not have the constraints necessary to resolve word sense ambiguities (as illustrated by the examples in Section 5.1 above).
- Quantitative measures will have to be *invented* for getting answers to quantitative queries. Current answers are truth-conditional and are insufficient for disambiguation (and coreference resolution) among candidate senses all of which are either acceptable or unacceptable according to truth-conditional answers.

5.3 Coreference experiments

This section contains brief accounts of different types of coreference problems and our findings in trying to answer the questions asked for in those cases using Cyc. We first discuss the problem of establishing coreference between any two expressions that are coreferring regardless of their form (cases 1 through 6) and, second, discuss the problem of establishing that any two expressions are not coreferring regardless of their form (cases 7 through 10).

⁹. One of the Spanish news articles being used currently in the Mikrokosmos project.

Case 1: PN acronyms

The first case concerns linking full proper nouns to acronyms (or even worse nicknames). For example, in the text we inspected, there is 1 potential instance in which "United" (or possibly "United Airlines") must be linked to "UAL". As it turns out, "United" and "UAL" are two different companies so the problem does not actually apply to this text, but in the MUC answer key, the items had been linked reflecting the importance of knowledge of the world.

One approach to this problem is to use inferencing on the basis of the information provided in the text coupled with a knowledge base and inferencing engine such as Cyc in order to establish the connection. Here, for instance, the question is whether "UAL" and "United" or "United Airlines" are being used to refer to the same entity. We know from the text such facts as:

UAL has a board of directors and a CEO,
United has a president,
the UAL board is concerned about the financial performance of United,
the strategy of the UAL CEO is to combine United with a hotel company and a car rental company,
the UAL board is satisfied with the financial performance the hotel company and the car rental company
etc.

If Cyc can reason that a company's board of directors is primarily concerned with that company's financial performance, or that a company's CEO has the power to merge that company with other companies, then it can decide that either (a) they are two different companies (correct), (b) they are one and the same company (incorrect), or (c) that there is insufficient information to say one way or the other (an okay result).

Findings

We first tried to find out if "board of directors" helps us. We have Business as the concept for "company" and BoardOfDirectors for "board of directors." However, there is no constraint on who can have a board of directors. Or, perhaps, we don't know how to express the above sense of "have" using Cyc's concepts. You could easily spend a day exploring Cyc and trying to find answers to sub-questions recursively and still not be able to answer the relevant questions one way or the other. In other words, there is no simple way to determine **what is not present in Cyc**. Knowledge representations with a better structure easily support such inferences.

Case 2: PNs referring to referents set up by definite descriptions

A second difficulty concerns identifying that a proper noun is being used to corefer to something that was set up by some other type of expression such as a definite description. In the text inspected, there are two examples. These come about when the author first refers to "the premier hotel company and car rental company", thereby establishing two new referents (or possibly a single new referent), and then, a sentence or two later, refers to "Westin Hotels" and "Hertz". The problem is to identify that "Westin Hotels" is being used to refer to the same entity that was referred to by "the premier hotel company" and that "Hertz" is being used to refer to the same entity that was referred to by "[the premier] car rental company". This connection can be made by knowing (or inferring) that:

Westin Hotels is a hotel company,
Hertz is a car rental company,
UAL owns a hotel company ("the premier hotel company")
UAL owns a car rental company ("[the premier] car rental company")

and then inferring that if UAL is satisfied with the earning of Westin Hotels and Hertz, then Westin Hotels must be the hotel company and Hertz must be the auto rental company that UAL owns. Cyc might be able to support this reasoning by providing some general rule to the effect that:

"if X owns Y, X wants Y to perform well"

and being able to apply that to organizations owning companies and therefore wanting those companies to have high earnings.

Equally important is to block establishing a coreference between "the Century Plaza Hotel", mentioned later in the text, and "the premier hotel company". From the text we know that:

UAL does not own the Century Plaza,
UAL manages the Century Plaza.

Using the same facts, then, Cyc might be able to establish that the Century Plaza is not "the premier hotel company".

Findings

We tried to answer the question about “if X owns Y, X wants Y to perform well.” I was able to find Owns for “own”, Desire and Need for “want,” but nothing for “perform well.” Tried every possible paraphrase we could think of with no success. Finally, I was able to find a rule that said if “an organization X has a subpart Y, then X has a VestedPositiveInterest in Y.” This seems close to what we were looking for, but does not really give us a Yes or No answer to the question above (because of all the other gaps to bridged between this assertion and what we are trying to ask, let alone what is in the text). This assertion alone will not help us figure out if Westin is the hotel and Hertz the rental company that UAL owns.

Case 3: pronominal anaphora disambiguation

To avoid a false positive match of "it" with "the executive committee" some general understanding of companies and (executive) committees appears to be needed. The reasoning is something along the lines of:

"it" is the possessor of a board of directors,
companies possess boards of directors,
"it" must be a company.

Therefore, add company to the pronouns other (morphological) constraints and follow the basic pronoun resolution procedure. This will rule out "the executive committee", which is not a company, as a possible coreferent (as well as "a member", also not a company).

Such inferencing is first triggered by establishing a semantic relationship between the pronoun and its local context ("its" is possessor of the following NP "board [of directors]"). The information about the other elements of the local context is then used to impose further constraints on the possible coreferents of the pronoun.

Findings

There is no information in Cyc on who can possess a board of directors. There is no relation between a company and a board of directors. Once again, I don't find any relation in Cyc between pilots (or employees in general) and companies. There is no way to infer that “it” must be a company because “it” employs pilots.

Case 4: CN coreference resolution--checking for semantic compatibility for CNs with anaphoric determiners

Certain common noun determiners, such as demonstratives (this, that, etc.) or certain quantifiers (each, all, either, neither, etc.), give the common noun an anaphoric "flavor" that we can take advantage of. Two examples from the "UAL" text are:

| | | |
|----------------------|-----|---------------------------------|
| that year | --> | 1985 |
| either side union | --> | United management the pilots' |

Following the pronoun resolution procedure, the strategy here is to inspect referents in reverse order of mention until a semantically compatible referent for the expression is found. There is no "form match" involved but we need to know the semantic category of the referent as provided by the common noun referring expression.

Thus, "that year" causes us to start looking at each referent, in reverse order of mention, until we find a year. To work, we need to know that "1985" is used to refer to a year, knowledge which could be supplied by Cyc.

With "either side", we first need to infer from "sides" that there has been some event referred to involving opposing groups. In the text we have been told that there has been a "litigation" in which "the 570 newly hired pilot" sued United to keep their jobs while "United management" claimed that they never had jobs because of a breach of pre-hiring conditions. This involved opposing groups, the 570 pilots and United management. As a result, we assume that "either side" is used to corefer to the 570 pilots or (inclusively) United management. If Cyc knows that litigations involve opposing groups, then it can be used to resolve the problem.

Findings

Cyc knows TheYear1984 and TheYear1986 but not TheYear1985.

Cyc knows that the word litigation means a LawSuit which is either a Trial or a LegalConflict. Assuming that we choose LegalConflict, we know that it is a Conflict. That is about it. Cyc has predicates for plaintiff and defendant but they are not in any way connected to the LegalConflict concepts. There is no information that a Conflict has two parties either.

Case 5: CN coreference resolution--checking for semantic compatibility for complements of CNs

In the following 2 examples:

the directors --> board members

the 570 individuals --> the "570 newly hired pilots"

the basic CN resolution procedure will fail because the head of the expression under consideration, "directors" or "individuals", does not match the head of the potential referring expression, "members" or "pilots". However, they both can be resolved by applying general, context independent inferencing. The reasoning is as follows. From the text we know that:

UAL has "a board of directors".

If Cyc can provide us with a general rule of the form:

if X is a member of Y and if Y is a group of Z, then X is a Z,

then "board members" are "directors" and, therefore, "the directors" is used to corefer to UAL board members. Similarly, we know from the text that:

there are 570 individuals who United was ordered to hire,

there are 570... pilots who received pre-hiring training.

If Cyc has information to the effect that "pilots" are people and that people are "individuals", then the "570 individuals" can be identified as coreferring to the "570 pilots".

Findings

Cyc doesn't tell us that a BoardOfDirectors is actually a group of directors.

Cyc knows that AirplanePilot is a Person. So, if you can establish that "individual" means Person, there should be no problem here.

Case 6: CN coreference resolution--checking for semantic compatibility of form-overlap PNs

For:

the airline --> United Airlines

the basic CN resolution procedure fails because "United Airlines" is not a common noun, that is, it is not included in the list of CNs that are being used by the resolution procedure to match against. We might consider adding PNs to the list of items being used to match against but it appears that that simple step will lead to more false positives than true positives (some example of this will come up in the discussion below). The "Airlines" in "United Airlines" after all does not refer to "an airline" but rather is part of the name of an airline.

Still, we might try matching against the PNs and, if successful, see whether inferencing can be applied to show that the referents are in fact the same and, therefore, the expressions are coreferring. In this case, the mention of "the airline" is in the following context:

Mr. Ferris and Jim Hartigan, president of United (=United Airlines), are taking steps to improve its (=United's) earnings.

First-quarter financial results of the airline(=??) should show a substantial improvement over....

Since "earnings" are part of "financial results" then "its (=United's) earnings" could be part of the "first-quarter financial results of the airline" but only if "the airline" is "United". So, while this begins by a CN to PN match to establish United as a possible referent of "the airline", inferencing is used to confirm that it is the referent. There are other ways to go about identifying this coreference relationship, however, that have nothing to do with the CN to PN match.

Findings

In Cyc, "earnings" means ReceivingPayment. There does not seem to be any link from there to "financial results."

Case 7: CN coreference filtering--checking for semantic compatibility for complements of CNs

This case concerns blocking the basic CN resolution procedure from identifying "comparable 1986 results" as being used to refer to the same results as the expression "first-quarter financial results" is being used to refer to. To do this we need to know that:

"first-quarter" refers to the first quarter of 1987

and then to show that:

the 1987 is not 1986

Therefore, there are two sets of results under discussion. Alternatively, we know from the text that:

the first-quarter results show a substantial improvement over the comparable 1986 results.

Thus, if Cyc knows that:

if X shows an improvement over Y, then X cannot be identical to Y,

it can be used to show that the two expression are not coreferring.

Findings

“quarter” can be mapped to CalendarQuarter in Cyc. I could not find any relation between CalendarQuarter and Year (or TheYear1987) in Cyc.

Yes, TheYear1987 and TheYear1986 are separate instances in Cyc.

The remaining query may be too difficult to formulate as a Cyc query.

Case 8: CN coreference filtering--checking for semantic compatibility for complements of CNs

Here, the basic CN resolution procedure must be blocked from identifying "the Journal's reference" in "the Journal's reference to the case of the “570 newly hired pilots”..." as coreferring to "the Journal's reference" in "the Journal's reference to Donald Trump's reputed desire...". The strategy for doing this is to recognize the case of the 570 pilots is not the same as Donald Trump's reputed desire and, therefore, that the references in question cannot be same. It is unlikely that Cyc can be expected to know that the case of the pilots is different from Donald Trump's desire since they are so entirely unrelated. But perhaps Cyc could show that:

cases (in general) are not the same as desires (in general).

If so, we can use Cyc to infer that the case of the pilots is not the same as Trump's desire and, thus, that the two references are not the same.

Findings

It is not likely that Cyc can infer this in any way other than by saying “any two things are not the same unless otherwise stated.”

Case 9: CN coreference filtering--checking for semantic compatibility for complements of CNs

This example requires blocking the basic CN resolution procedure from identifying "the outside directors" as being used to corefer to the referent of "the directors", i.e., the members of the UAL board of directors. This one is somewhat complicated by the fact that the outside directors are members of the UAL board. Here, it seems to me that if Cyc knows that, or is able to reason that:

boards of directors consist of inside? directors and outside directors,

the outside directors are NOT the entire board,

then we can use it to conclude that "the outside directors" cannot be referring to the same thing as "the directors" (provided, of course, we assume the "the directors" refers to ALL the directors).

Findings

As mentioned earlier with reference to another case, there is no information in Cyc about the members of a BoardOfDirectors.

Case 10: CN coreference filtering--checking for semantic compatibility for complements of CNs

The final case involves several instances of potential coreferring although only two referents, the pilots of United Airlines and the 570 pilots that United trained in 1986 and eventually hired in 1987. There are 6, or possibly 7, referring expressions involved and 12, or possibly 16, cases of blocking required.

The first occurrence of the problems involves the expression "the “570 newly hired pilots”" which might potentially be used to corefer to the United Airlines pilots which were initially referred to by "the pilots" in the expression "the pilots’ proposal to...". To block this coreference we need to reasoning that:

not all United pilots have been recently hired,

Therefore, "the 570 newly hired pilots" could not possibly be referring to the same group of people as "the pilots" was used to refer to and, thus, we must set up a new referent, the 570 pilots.

The second occurrence of the problem involves the expression "pilots" in "these are pilots who...". This is, in fact, only a possible occurrence in that it could be argued that "pilots" here is being used attributively and therefore the basic CN resolution strategy should be blocked from applying (and would be if we have marked constructions such as "X is Y" as implying X is some particular individual and Y is some conceptual category). However, assuming that "pilots" is coreferring, one approach to solving the problem would be to first establish that "pilots" and "these" are coreferring by virtue of being connected by "be" and then rely on having established that "these" is coreferring with "the 570... pilots" by the pronoun resolution procedure.

An alternative approach is to extract from the complement of "pilots" the fact that:

they would be expected to fly if United's service were interrupted by a strike,

and then reason that:

a strike means that the employees, and specifically the pilots, do not perform their work, specifically fly,
management expects this to be the case,
management DOES expect these pilots to fly,

therefore, "these pilots" cannot be the United pilots.

The third and fourth occurrences of the problem involves the expression "the pilots" in "the pilots' strike". The basic CN resolution procedure must be blocked from identifying this as coreferring to the 570 pilots thus far referred to by "the "570 newly hired pilots"" and, possibly, "pilots" in "these are pilots who...". This can be done by a subpart of the reasoning described in the previous paragraph, namely:

if there is a strike by pilots then the pilots on strike are employees of the company,
management does not expect striking pilots to fly.

Therefore, "the pilots" must refer to the United pilots and not the 570 pilots.

Occurrences five through eight of the problem involve the expression "the 570 pilots" used twice to refer to the 570 pilots. The basic CN resolution procedure must be blocked

from identifying them as coreferring to the United pilots, already referred to by "the pilots" in "the pilots' strike" and in "the pilots' proposal". Here the reasoning would be:

there were pilots flying for United before the 570 were "pre-hired",
therefore there must be more than 570 United pilots.

Thus, these expressions cannot be used to corefer to the United pilots.

Occurrences nine through 12 involve the expression "the pilots" in "the pilots' union", again, being used to refer to the United Airlines pilots. The basic CN resolution procedure must be prevented from identifying the expression as coreferring to the 570 pilots, at this point referred to by "the 570 pilots" on two occasions, "the "570 newly hired pilots" and, possibly, "pilots" in "these are pilots who...". Here the reasoning is that:

a pilots' union is an organization for all pilots employed by an
employer
the 570 pilots are not all the pilots employed by United,
(indeed may not even be employed by United).

Therefore, "the pilots" must be referring to the United pilots not the 570 pilots.

The final four occurrence of the problem involves the expression "our pilots" which is used to refer to the United pilots generally. The basic CN resolution strategy must be blocked from identifying the expression as being used to refer to the 570 pilots which have been referred to at this point by "the 570 pilots" on two occasions, "the "570 newly hired pilots" and, possibly, "pilots" in "these are pilots who...". Here we know "our" corefers to the board of directors of UAL and so the expression in question is really "the pilots of United Airlines" which can be blocked using the final steps of the reasoning described in the previous paragraph.

the 570 pilots are not all the pilots employed by United,
(indeed may not even be employed by United).

Therefore, "the pilots" must be referring to the United pilots not the 570 pilots.

Findings

This case was rather detailed. We didn't go into the details. However, we expect to find some relevant concepts in Cyc and still not be able to make the required inferences.

5.4 Summary: Types of queries

We summarize our findings by revisiting the types of queries needed for disambiguation and other tasks and saying whether or not Cyc can answer each of the types.

Taxonomic queries: The simplest type of query is one where the two concepts are linked through a chain of taxonomic links (i.e., one is an ancestor of another). These are the cases where the constraints **are** met literally. Cyc can easily answer such queries.

Taxonomic queries are also examples of known-predicate queries since the predicate `#$gen1s` is given to Cyc. Another simple case is when the two concepts are linked directly (i.e., through a single predicate). Cyc can answer such queries provided we know the “arity” of the predicate. For example, if we know that it is a binary predicate (i.e., does not take any argument other than `Concept1` and `Concept2`), we can ask:

```
(#$LogAnd (#$instanceOf $X #$BinaryPredicate)
           ($X #$Concept1 #$Concept2))
```

i.e., What are the `BinaryPredicates` that relate `Concept1` and `Concept2`?

The query is complicated because of the need to constrain the unknown predicate to the class of `BinaryPredicates`. Otherwise, Cyc does not process the query since it expects such unrestricted queries to be prohibitively expensive. Of course, we may not wish to constrain the search to just `BinaryPredicates`. We could repeat the query, if necessary, for ternary predicates, and so on. Moreover, *queries of the above type seem to be very expensive to answer in Cyc*. This is because the search space is the entire set of `BinaryPredicates` instead of being just those that link `Concept1` to other concepts. `Concept1` may have only a few links to other concepts (or a few properties, say, 1-25) but there are over one thousand `BinaryPredicates` in the Cyc knowledge base.

If there is indirection in the link between `Concept1` and `Concept2`, even when the same predicate is used throughout the chain of links, we need to know the number of links (or intermediate concepts). For example, if we know (how?) that there is just one intermediate node, we could ask a second-order query such as:

```
(#$LogAnd (#$instanceOf $X #$BinaryPredicate)
           ($X #$Concept1 $Y)
           ($X $Y #$Concept2))
```

i.e., What are the `BinaryPredicates` that relate `Concept1` to `Concept2` through an intermediate concept?

We need to repeat the query for each number of possible intermediate links. This means, it is not practical to ask Cyc such queries except in the case of direct links. There is also the problem of Cyc finding many short, irrelevant answers to second-order queries such as the above (e.g., that both Concept1 Concept2 are instances of Thing). Excluding meaningless paths is, however, a central problem in knowledge-based word sense disambiguation. We (CRL and DoD, under a separate IDEAS project) are working on this problem using semantic regularities in languages such as the kinds of metonymies and metaphors permitted in a language.

However, in the most interesting cases (from the point of view of sense disambiguation), the two concepts will only be related through a path that has more than one type of link (or predicate). Except in the case of taxonomically related concepts, the predicate(s) will be unknown in the query.

A general problem with these queries is that most Cyc predicates are at the instance level. That is, they check for the existence of instances of concepts and relationships between instances. This seems to unnecessarily add to the complexity of queries. Most relationships between concepts hold no matter whether they have instances in a particular context; they are best represented at the concept level. For example, the constraint that *what is eaten in an eating event must be food* can be stated in either of the following ways:

```
for every EatingEvent, there exists an instance of Food
that is the objectActedOn.
```

```
for every EatingEvent, if there is an objectActedOn, then
that object is an instance of Food.
```

As already noted (Section 5.1.2), Cyc seems to have chosen the former representation which is unnecessary. It also makes logical inferencing more complex by introducing unnecessary instantiation (i.e., skolemization).

Complex queries such as the following always return a NIL answer (with a “reason” also NIL).

```
(#$LogAnd (#$InstanceOf $X #$BinaryPredicate)
           ($X #$TeethCleaning $Y)
           (#$InstanceOf $Z #$BinaryPredicate)
           ($Z $Y #$AnimalActivity))
```

i.e., What are the pairs of BinaryPredicates that link TeethCleaning to AnimalActivity through an intermediate concept?

As already noted, such arbitrary-path queries are essential for knowledge-based NLP. One basic requirement for answering such queries seems to be the ability to find all the links from a given concept to other concepts in the knowledge base. Representations such as semantic networks (or network of frames) are designed for finding such relationships. In Cyc, however, it does not seem to be possible to find all the predicates that link a concept to other concepts in the knowledge base (either explicitly present or inherited or inferred). There does not seem to be a general solution to this problem that covers all of the necessary types of queries (listed in Section 4.2). The main problem seems to be Cyc's knowledge representation based on predicate logic. The representation in Cyc is not *object-oriented*, not in the sense of programming languages, but, in that all of the knowledge about a particular concept is not represented in one place, and hence, is not accessible in a tractable manner.

An experimental path-finding program was developed by the Cyc group. We did not have the resources to test this software to determine if it is fast enough in doing the search and effective enough in aiding us filter irrelevant and meaningless paths. However, such a search does not seem to be tractable in Cyc, given its logical representation. For example, the search space is determined by the total number of (binary, say) predicates in Cyc which is much higher than the total number of inter-concept relations that a given concept has. Although microtheories in Cyc (Lenat et al., 1990) are meant to constrain the search, for applications in NL semantics, it is unlikely that a real-world text allows us to determine which microtheory in Cyc the text is about and exclude all other microtheories from consideration.

The Cyc group informed us (Burns, personal communication) that there are low-level indexing functions, not part of the functional interface to Cyc, that can perform more powerful search operations in the knowledge base. However, we do not see how the functions can overcome the lack of structure in the representation. For example, they suggest looking for axioms in which both concepts co-occur to find possible paths from one to the other. However, as illustrated earlier in this document, this method covers only direct relationships between concepts, not arbitrary path queries with several, unknown, intermediate concepts. Even if such search operations can be constructed using the said low-level indexing functions, they are likely to be prohibitively expensive given the unstructured, "sea of assertions" representation of knowledge in Cyc.

Moreover, Cyc's answers to queries are binary, truth-conditional values (plus any variable bindings). It does not give us estimates for how true a given statement is. Such information is necessary for comparing different choices and selecting the best combination of word senses for a sentence. It may be possible to devise quantitative measures perhaps based on the proof process to obtain numbers along with truth-conditional answers. However, the structure and organization of knowledge in Cyc's representations do not inherently provide quantitative answers.

In summary,

- Taxonomic queries are easily answered;
- Direct-link queries are answered if the relationship (link) concerned is known. Cyc can find unknown links by searching among all predicates of a given arity (e.g., binary predicates) and given ordering of their arguments.
- Arbitrary path queries: Cyc does not seem to be capable of finding such paths.
- Quantitative queries: Cyc does not seem to have cost information represented explicitly and does not seem to provide answers other than true or false.
- Constrained-path queries may be possible by Cyc only when the types of all links in the pattern are known beforehand.

6. Discussion and Conclusions

Our overall assessment of the content and usability of knowledge in Cyc for NLP can be stated as follows:

- Cyc has a large amount of knowledge that is potentially very useful for NLP; however,
- Cyc's knowledge is not readily applicable to problems such as word sense and coreference resolution, either in an exclusively Cyc-based system or one where Cyc is used to improve the performance of other methods.

Some of the main reasons why Cyc's knowledge is not readily applicable to NLP include:

- the absence of certain kinds of knowledge necessary for NLP; in particular, the absence of selectional constraints on relations between concepts;
- incomplete and non-uniform coverage of concepts and knowledge about individual concepts (in terms of depth and breadth of coverage);
- inability to access everything known about a given concept (both explicit and inherited or inferred knowledge); and
- inaccessibility (or inefficient accessibility) of existing knowledge due to the lack of structure in the "sea of assertions" representation.

In the rest of this section, we discuss some issues in the acquisition and representation of knowledge in Cyc. These discussions are rather brief. We will be happy to elaborate on any of them if either DoD or Cycorp is interested.

6.1 Acquisition Methodology

The driving force behind knowledge acquisition in Cyc is not clear. For example, it is not clear whether Cyc's coverage should be measured against a particular task or application or against a standard resource such as a dictionary or encyclopedia. However, we believe

strongly that a well-defined driving force is essential for achieving uniform coverage at all levels, namely, ontological coverage of concepts, attributes and relations of a concept, and coverage of lexical items in a language (i.e., English in the case of Cyc).

Our experience in ontology development and knowledge acquisition in the Mikrokosmos project (Mahesh, 1996) seems to indicate that *situated acquisition* (Mahesh and Nirenburg, 1995) is the key to attaining uniform coverage and good quality and utility of the knowledge acquired. By situated acquisition we mean a methodology in which:

- a task external to the knowledge acquisition process drives the acquisition; for example, building a semantic lexicon drives ontology acquisition in the Mikrokosmos project.
- most ontological decisions are made by negotiation with ontological customers, namely, those who use the acquired knowledge for lexicon development, machine translation, planning, or other application tasks.
- knowledge acquired is immediately used and tested by ontological customers, thereby providing immediate feedback and quality checks on each piece of knowledge.

It is not clear that the acquisition of Cyc was situated in such an environment. Although Mikrokosmos and other similar projects have invested far fewer person-years of effort in knowledge acquisition than Cyc, some of the lessons from these projects may benefit Cyc, for example, in attaining uniform ontological coverage.

6.2 Knowledge Representation and Ontological Issues

It appears that many of the problems in applying Cyc's knowledge for disambiguation and coreference resolution stem from the underlying knowledge representation in Cyc. It is our hypothesis that Cyc's representation is not designed to enable efficient access of *all the knowledge about a given concept*. It is clear to us that the hypothesis is true given the functional interface to Cyc (the FI or the Web browser interfaces). Whether the problem can be solved by providing new functionality in the interface or by augmenting Cyc's indexing schemes is to be determined empirically.

Cyc's Representations: Lack of Structure?

Our experience in knowledge acquisition (e.g., Mahesh, 1996) has shown that:

- knowledge represented locally for any given concept is not sufficient either for using that concept in an application or for acquiring further knowledge (about the same concept or other concepts in the ontology); we must be able to see and quickly digest all of the *inherited or otherwise inferred knowledge* about the concept. For example,

how can one add a new axiom involving a concept without seeing what is already known about the concept?

- it is not practical to leave the burden of obtaining inherited or inferred knowledge to the human user; for example, although, given modern interfaces such as Cyc's web interface, it is easy to go up a hierarchy and examine knowledge represented locally at each of the ancestors of a concept, it is not practical to expect the user to figure out which of these pieces is legally inherited by the concept of interest. It is certainly not practical to expect the user or acquirer to be already entirely familiar with (even portions of) a large knowledge base such as Cyc.

We have also demonstrated that accessing all the knowledge about a concept is essential for answering arbitrary-path queries in disambiguation or coreference resolution. We therefore believe that the underlying representation of knowledge must support efficient ways of retrieving everything known about a concept: locally stated, inherited, as well as otherwise inferred knowledge. Why is it difficult for Cyc to provide such functionality? The answer seems to lie in the design of its knowledge representation. Cyc switched, we believe, from its original frame-based representation to a less structured logic-based representation, in order to increase the expressive power and avoid some of the ad-hoc representational kludges inevitable with simple frame-hierarchy based representations (Whitten, et al, Version 2.0). As a result of this, Cyc appears to suffer from the above drawback. In Cyc, knowledge about a concept is spread throughout in individual assertions in a vast "sea of assertions," without any apparent organization that brings it all together in one place (i.e., in a frame or object).

7. Proposals for Further Work

This section states briefly some of our thoughts on potential collaborations and further work. We will be happy to elaborate further on any of the following.

7.1 Towards a High Performance Knowledge Base

Some recommendations for building a high performance knowledge base:

1. Knowledge acquisition should follow a clear statement of goals:

- What must this KB support? Tasks and situations where it will be used must be defined in advance. E.g., coreference resolution for MT and planning for scheduling and logistics in such and such situations.
- Example problems and scenarios must be built and types of queries that the KB must answer be analyzed.
- Types of knowledge to be included in the KB must be determined in advance.

- Level of coverage (depth, breadth, etc., see Nirenburg et al., 1996) to be attained for each type of knowledge must be determined.
- Testing methodology for ensuring uniform coverage, utility for the chosen tasks, and reasonable degree of correctness must be developed in advance.

2. Should we acquire from scratch or build on existing ontologies?

We should try to merge several existing ontologies and build from them because:

- all general-purpose ontologies built so far are remarkably similar to one another
- it seems possible to merge the bulk of these ontologies which happen to be in the middle levels in their taxonomies; this appears to be much less expensive than building from scratch (e.g., Hovy, Presentation at the September 1996 meeting of the Ontology Standards ad hoc ANSI committee).
- it is not reasonable to shoot for a single, unique, correct ontology; we should plan to allow multiple top-level classifications to be superimposed on the KB. Ontologies differ a lot in their top-level classifications. However, it appears that the top few levels of the classification are not as critical as the lower-level distinctions in solving particular problems in real tasks.
- similarly, the High-Performance KB should provide for plugging in domain-specific ontologies at the lowermost levels. E.g., the entire library classification system can be imported and plugged in under the FIELD-OF-STUDY concept in the Mikrokosmos ontology (Mahesh, 1996).

Moreover, *what evidence do we have to believe that building from scratch will result in an ontology of significantly better quality than previous ones (e.g., Cyc)?*

The work of the Ad-Hoc ANSI Committee on Ontology Standards is particularly relevant in this context. Current plans of the committee includes the development of a Reference Ontology by merging the top and middle levels of Cyc, WordNet (Miller, 1990), Pangloss (Nirenburg, ed., 1994; Knight and Luk, 1994), Mikrokosmos (Mahesh, 1996), and EDR (EDR, 1991) ontologies.

7.2 Cyc Knowledge Acquisition Tools

CRL could build corpus-based software assistants to aid knowledge acquisition and quality improvements in Cyc. Such an assistant could use Cyc's existing lexicon to process a corpus and suggest missing relations and constraints between Cyc's concepts as well as ontological gaps to filled by acquiring new concepts.

7.3 Quantitative Evaluation of Cyc

The assessment reported here could be followed by a more extensive, quantitative, empirical evaluation of Cyc for specific NLP applications. For example, we could adopt the Mikrokosmos system to use Cyc for Spanish-English machine translation providing a basis for comparing the utility of Cyc's knowledge with that of Mikrokosmos' own knowledge which, unlike Cyc, was acquired expressly for machine translation.

8. Acknowledgments

This work was supported by the United States Department of Defense under contract MDA904-92-C-5189. We would like to thank Cycorp, Inc. for providing copies of latest versions of the Cyc knowledge base, software, and documentation. We would also like to thank Kathy Burns and Karen Pittman of Cycorp for explaining Cyc to us in detail and for answering our many questions promptly. We also thank Kathy Burns and Mickey Duniho for their many useful comments on initial drafts of this report and Evelyne Viegas for her input on lexical issues. We thank Adam Cable for installing Cyc at CRL and Praveen Mamnani for figuring out how to use Cyc.

9. Bibliography

Beale, S., Nirenburg, S., and Mahesh, K. 1995. Semantic Analysis in the Mikrokosmos Machine Translation Project. In the *Proceedings of the Second Symposium on Natural Language Processing (SNLP-95)*, August 2-4, 1995, Bangkok, Thailand.

EDR. 1991. *Proceedings of the International Workshop on Electronic Dictionaries*, EDR TR-031, Japan Electronic Dictionary Research Institute, Tokyo, Japan.

Fass, D. 1988. An account of coherence, semantic relations, metonymy, and lexical ambiguity resolution. In *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence*. ed. S. I. Small, G. W. Cottrell, and M. K. Tanenhaus. Morgan Kaufmann Publishers.

Knight, K. and Luk, S. K. 1994. Building a Large-Scale Knowledge Base for Machine Translation. In *Proc. Twelfth National Conf. on Artificial Intelligence, (AAAI-94)*.

Lenat, D. B. and Guha, R. V. 1990. *Building Large Knowledge-Based Systems*. Reading, MA: Addison-Wesley.

- Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. 1990. Cyc: Toward programs with common sense. *Communications of ACM* 33, no. 8 (August 1990).
- Mahesh, K. 1996. Ontology Development: Ideology and Methodology. Technical Report MCCS-96-292, Computing Research Laboratory, New Mexico State University.
- Mahesh, K. and Nirenburg, S. 1995. A situated ontology for practical NLP. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.
- Mahesh, K. and Nirenburg, S. 1996. Meaning Representation for Knowledge Sharing in Practical Machine Translation. In *Proc. FLAIRS-96 special track on Information Interchange*, Florida AI Research Symposium, Key West, FL, May 19-22, 1996.
- Miller, G. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography* 3(4) (Special Issue).
- MUC6. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC6)*, DARPA, Morgan-Kaufmann, Nov. 1995.
- Nirenburg, S. editor. 1994. The PANGLOSS Mark III Machine Translation System. A Joint Technical Report by NMSU CRL, USC ISI and CMU CMT, Jan. 1994.
- Nirenburg, S., Carbonell, J., Tomita, M., and Goodman, K. 1992. *Machine Translation: A Knowledge-Based Approach*. Morgan Kaufmann Publishers, San Mateo, CA.
- Nirenburg, S., Mahesh, K., and Beale, S. 1996. Measuring semantic coverage. In the *Proceedings of the International Conference on Computational Linguistics, Coling-96*, August 5-9, 1996, Copenhagen, Denmark.
- Nirenburg, S., Raskin, V. and Onyshkevych, B. 1995. Apologiae Ontologiae. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation*. Leuven, Belgium, July.
- Onyshkevych, B. A. 1995. A generalized lexical-semantics-driven approach to semantic analysis. Dissertation proposal. Program in Computational Linguistics, Carnegie Mellon University.
- Whitten, D. et al., Version 2.0. The Unofficial, Unauthorized CYC Frequently Asked Questions Information Sheet, Version 2.0. Posted regularly on the comp.ai, comp.ai.philosophy, and comp.ai.nat-lang news groups on the Internet. Copyright by David Whitten and for some parts by MCC and Cycorp.

