# Using Open Source Tools in Text Mining Research

**M. Keith Wright**

Decision Sciences Consulting Group
7701 Montague Dr.
Austin, TX 78729
(512) 249-0059
keith.wright1@worldnet.att.net

## ABSTRACT

*To date there have been notable commercial successes of the open source software development model, such as Linux, Apache, and Open Office. However, the impact of open source tools on research and teaching is less clear. Some researchers now argue that the open source software development model, combined with the internet, is poised to contribute to rapid advances in a new generation of intelligent text mining systems. This paper examines the likelihood of that scenario, and suggests further research opportunities in that area. In particular, this paper compares the usefulness of the following open source products for research in text mining: openCyc, WordNet, and ConceptNet.*

## INTRODUCTION

### *Opensource software*

The open source *movement* began in 1998 as an outgrowth of the *Free Software Foundation* (FSF), whose literature explains that it is the software license, not the price, that determines *freedom (http://www.gnu.org/). Free* licenses convey permission for anyone to use, copy, and distribute the software -- either verbatim or with modifications --- either gratis or for a fee; and stipulate that the source code is available. If a component is *free*, then anyone has permission to include that component in a *free* operating system -- like GNU/Linux. The *FSF* literature says *free* software should be a community project, and that its dependents should contribute new product back to the community.

The most common open source license is the *GNU* general public license. This license is sometimes referred to euphemistically as a *copyleft* license.

However, the term *free software* is a term that business people may not feel comfortable with, because it may connote inferior quality. This thinking resulted in the *Open Source Organization, which* has published a formal definition of an *open source* license. Although this definition is actually more restrictive than the GPL license, most people use the terms *freely available* software and *open source* software, interchangeably. This terminology change has proved effective, because many businesses now use open source software.

The fundamental difference between the *free* and *open* software communities is in their values. You can think of *open source* as a development methodology; and *free software* as a social movement.

Last year, upon leaving IBM after several years, I became interested in the breadth of software titles available price free over the internet. I estimated that there are over 30,000 of these. The stated purposes of many of these tittles are research and teaching. Some of the most interesting are those that attempt to read natural language text.

Reading natural language text requires much back ground knowledge -- the kind of knowledge not evidenced in ordinary human discourse. Systems that seem to have this kind of knowledge are said to be *intelligent.* Most people call this type of knowledge *common sense*.

## *COMMON SENSE SYSTEMS*

Marvin Minsky, once estimated that commonsense knowledge is composed of about forty million facts about the world we live in (Minsky, 1987). These are facts so ordinary that they are usually excluded from everyday conversation. For example:

- *You have to be awake to eat.*
- *You can usually see people's noses, but not their hearts.*
- *Given two professions, either one is a specialization of the other or else they are likely to be independent of one another.*
- *You cannot remember events that have not happened yet.*
- *If you cut a lump of peanut butter in half, each half is also a lump of peanut butter; but if you cut a table in half, neither half is a table.*

Such assertions are unlikely to be published in textbooks, dictionaries, magazines, or encyclopedias; even those designed for children.

Contemporary information systems, such as Google for example, use a combination of keyword-spotting, synonym generation, syntax parsing, and mathematical methods to understand search queries. While Google is certainly impressive, it delivers only semantically shallow results. For example, when I presented the following search string to Google – "List of under valued companies" -- I got back **215,000** hits, the top four of which were interesting, but not exactly what I asked for. (See table 1.)

| *List of under valued companies* |
|---|
| http://www.Hoovers.com |
| http://www.harrisinfo.com |
| www.gabelli.com/Template/fund**list**.cfm.financialsense.com/fsu/posts/dancy/reviews/072804.html |
| www.stockhouse.ca/featuredsector/ index.asp?sectorname=**undervalued** |

**Table 1: Google Results**

Perhaps some common sense rules would have helped Google here. For example, "An item in a list is not the same as an agent that performs lists."

## LITERATURE REVIEW AND CASE STUDIES

Information systems have made stunning advances in the last twenty years. They can now successfully diagnose medical symptoms, analyze and repair problems in space craft, plan travel routes, and transcribe speech into text. We now have programs that can play chess at the level of the best players, solve complex logistics problems, and bring the largest libraries in the world to our finger tips. Each of these successful information systems employs a different reasoning method. Despite these varieties of successful reasoning methods, we have not yet built a machine that can read the simplest children's story and answer questions about it.

Just as each of today's successful information systems employs a different type of reasoning method, each bit of common sense likely requires its own specialized methods of reasoning. Today, MIT researchers believe that giving computers common sense is not about making some particular reasoning method work, it is about making systems that contain many types of knowledge and many ways of reasoning (Minsky, 1987).

The scale of these problems has been terribly discouraging. Those few who have tried have found that you need a tremendous amount of diverse knowledge to understand the simplest children's story. This is a problem as large as any other in computer science to date. The first major attempt to manually build a database of human common sense knowledge was the *Cyc* project.

### *CYC*

The *Cyc* project began in 1984 as part of the Microelectronics and Computer Technology Corporation, and has spent over fifteen years manually entering commonsense knowledge into a Lisp-like logical framework (Lenat,1995). The original scope of the project was to give a computer all the necessary background knowledge that a human uses to read an en*cyc*lopedia. Hence the name *Cyc*. Today *Cyc* has a knowledge base of over 1.6 million facts, concepts, and relationships.

*Cyc* is offered in two different packages. The full *Cyc* system is now a commercial product of CycCorp Corporation in Austin, Texas (http://www.cyc.com/). There is also a freely available version available at http://www.opencyc.org/. CycCorp also purportedly offers a non-public freely available version called *researchCyc* (De Oliveira, J, 2004).

*Cyc* is organized into an *Upper Ontology* and several *micro-theories* (figure 1). The upper ontology contains very general concepts, while each micro-theory contains specific domain knowledge. The upper ontology consists of assertions about space, time, causality, and human behavior. Spatial relations include *Surfaces, Portals, Cavities, Shapes, Arcs, Linear-Planar, RoundShape, and Amorphous.* Direction and Orientation predicates include *inFrontOf-Directly* ,and *inFrontOf-Generally*.

There are also 120 assertions about emotions such as *Abhorrence* , *Adulation, Relaxation,* and *Gratitude* . Furthermore there are assertions about relations between emotions such as

*ContraryFeelings*, *FeelsTowardsObject* , *AppropriateEmotion*, *FeelsTowardsPersonType*, and *ActionExpressesFeeling.*

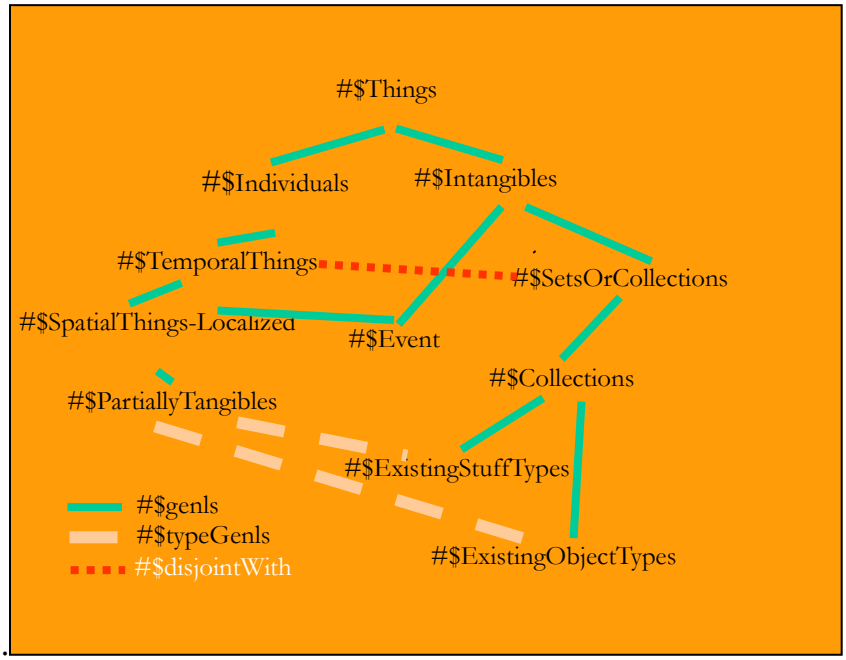


**Figure 1: *Cyc* Upper Ontology**

Events are represented by 37,000 predicates that relate actors to event instances. Over 200 *Roles* are performed during event instances, and *Actors* perform roles.

*Cyc's Micro-Theories* include those about biology, chemistry, food, weather, geography, and a few others.

While the *Cyc* project has done a commendable job of organizing a large chunk of scientific knowledge into a database that a logic inference engine can navigate, *Cyc* lacks user friendly knowledge capture tools.

For example, if a zoologist researcher wanted to build an information system based on *Cyc* he/she would first have to understand a large piece of the *Cyc* ontology, and then learn how extend it with knowledge from the specific problem domain, zoology in this case. As an example of this complexity, consider that the concept *dog* in *Cyc* is an instance of more than *ten* other concepts: *OrganismClassificationType*, *CanineAnimal*, *Object*, *Collection*, *CanineAnimal*, *FrontAndBackSided Animal*, *BilateralObject*, *BiologicalLivingObject*, *AirBreathingVertebrate*, *NaturalTangibleStuff* , and *NonPersonAnimal*. Thus, the zoologist would have to decide where and how many places to put his new concept.

Fortunately, *openCyc*, comes with extensive installation instructions and tutorials. I tried release 0.7 on my XP laptop, and it was easy to install. You simply unzip the download file, and execute the file openCyc.http.bat from a command line window. After about thirty seconds, in the

command window you will see the command prompt CYC(1). Altought I didn't try the command line interface (It requires skills in the LISP language.)

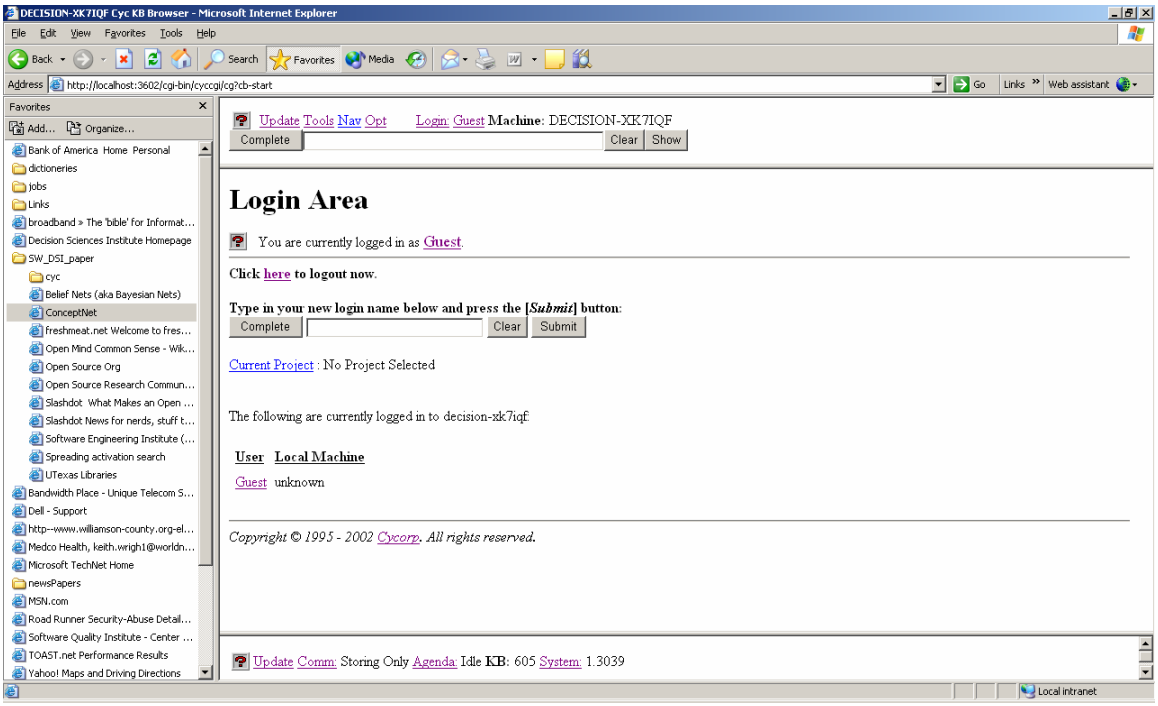


**Figure 2: The *OpenCyc* Knowledge Browser**

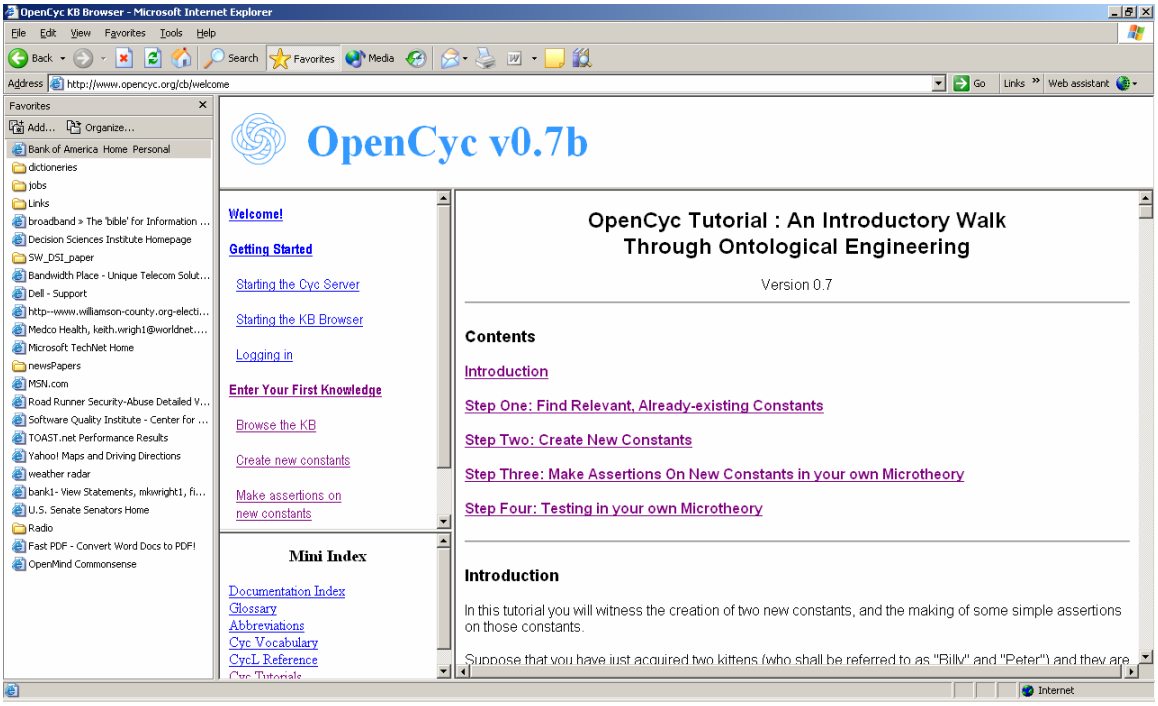


**Figure 3: The *openCyc* Documentation**

when you get that prompt, you then you can point a web browser as show in figure 2. (*openCyc* comes with its own local HTTP server.) In a separate browser instance, you can access the tutorial as shown in figure 3, by clicking *enter your first knowledge*.

This will bring up the extensive tutorial, which I tried, and will now discuss.

***From the Opencyc Tutorial***

The tutorial presupposes that the problem domain is animal behavior. You have just acquired two kittens, *Billy* and *Peter*, and you want to make sure these two concepts are adequately covered in the *openCyc* knowledge base. Here's how you'd go about adding those two concepts correctly and effectively into the *openCyc* knowledge base. It is surprisingly difficult.

First of all you should look for relevant existing *constants* to use to make assertions about Billy and Peter. A common mistake is to create a new constant you think fills a hole in the ontology, only to find an equivalent lurking under a name you did not expect.

You will want to assert that Billy and Peter are kittens. So you'd enter the word *kitten* in the text field next to the *Complete* button in the top frame in figure 2, and click on the *Show* button.

You will find no constant called #$Kitten in the *openCyc* knowledge base. You won't find *cat* either. So your next step will be to try something more general, like *pet*. You will find that *Cyc* has two such constants: #$DomesticPet, and #$NonPersonAnimal. Further digging will reveal that #$DomesticPet is a more specific constant than #$NonPersonAnimal, since #$DomesticPet is a specialization of #$DomesticatedAnimal, which is a specialization of #$TameAnimal, which is a specialization of #$NonPersonAnimal. Note that more specific *collections* inherit and specialize the information that exists on the collections they are specializations of. (This hierarchy is ordered using the predicate #$genls.)

#$DomesticPet is therefore better because, in rule based knowledge engineering, it is best to be as general as possible. #$DomesticPet is a *collection* so, to use this constant to make assertions about Billy and Peter, you'll have to assert that they are both members of this collection.

To assert that Billy and Peter are brothers you will need to keep in mind they are also brothers of each other, and that such relationships between two or more things are handled in *Cyc* using *predicates* rather than *collections*. So then you'd need to confirm there is not a predicate called #$brothers, and then try something more general, like #$relatives, which does exist in the *Cyc* knowledge base. But perhaps there is something a bit more specific, if not as specific as #$brothers. So, from the *2 possible terms* page, you'd click on the *relatives* link to view the details of #$relatives . Then you'd look at the *Term Index Frame*, under Arg2, and click on *genlPred*s. Now you'd see list of all the predicates which are direct specializations of #$relatives (See table 2.). So you'd see that #$siblings is more specific than #$relatives, and it is also true of Billy and Peter. So you could use #$siblings.

In addition you might want to assert that Peter and Billy *like* each other. So you'd need to see if there is some term (probably a predicate) you can use. You'd see constant called #$likes-

Generic, and you'd need to know that this a general term designed only as a knowledge organizer, not for inferencing. So you'd then need to look for something more specific. For this you need to find the appropriate #$specPred. You'd find one called #$likesObject, which means that when a sentient agent interacts with an object, the agent feels enjoyment. The kinds of interactions that produce enjoyment depend on the type of object. Thus, "Joe #$likesObject the Mona Lisa" implies that Joe feels enjoyment when viewing the Mona Lisa; and "Joe #$likesObj pizza" implies that Joe feels enjoyment when eating that kind of food. There are some specialized predicates of #$likesObject that give more information about the kind of interaction between the agent and the object results in enjoyment -- #$likesSensorially and #$likesAsFriend.

| |
|---|
| genlPreds grandchildren relatives) |
| (genlPreds cohabitingFamilyMembers relatives) |
| (genlPreds children relatives) |
| (genlPreds coreRelatives relatives) |
| (genlPreds siblings relatives) |
| (genlPreds biologicalRelatives relatives) |

**Table 2: Specializations of #$relatives**

So then you'd probably not be sure that you'd want to assert #$likesObject of Billy and Peter, because Billy doesn't quite like Peter in the way that Joe likes pizza. Further digging will reveal the predicate #$likesAsFriend, and that this predicate implies that there may be romantic feelings, and other feelings as well. To be sure, you should check what are called the *argument-constraints* on #$likesAsFriend. If this predicate is intended only to apply to human beings, then next to the #$arg1Isa and #$arg2Isa predicates in the main frame for #$likesAsFriend (assuming it has been well-ontologized) #$Person should appear. Further digging reveals that #$Person does not appear there, only #$PerceptualAgent. Will Billy and Peter meet this criterion? Yes they will, once they have been asserted to be members of the collection #$DomesticPet , because (#$genls #$DomesticPet #$PerceptualAgent) is true. So you should use #$likesAsFriend.

Asserting how old Billy and Peter is even more difficult than what you have seen so far. However, I proceeded successfully through the tutorial until it came time for the actual assertions into the knowledge base. At this point the *openCyc* HTTP server hung, and I could go no further. Sending emails to published addresses produced no help. However I have since discovered that there is a small but active *Cyc* users group here in Austin, Texas, the home of CycCorp, but membership is by invitation only (De Oliveira, J, 2004). From the users group I learned that version 1.0 of *openCyc* might fix my problem. It is scheduled for sometime in 2005.

*Cyc*'s creators claim that *Cyc* excels in deductive reasoning and situations which can be posed unambiguously (Lenat, 1995). But to use *Cyc* to reason about arbitrary text, you must follow a procedure similar to what we did in the above tutorial which maps new concepts into *openCyc*'s proprietary logical representation, described by its own Lisp-like language *CycL*. Although there is an HTML based tool to help, you have just to read the tutorial to see how complex the process remains.

From my experience with *openCyc* tools, I can not recommend them for independent academic research projects. However, reading through the extensive freely available documentation is very educational. CycCorp admits that the current *Cyc* knowledge base is still one or two orders of magnitude away from what is needed for common sense (De Oliveira, J, 2004). A million and a half pieces of knowledge is still terribly far away from the forty million that Minsky speculates is needed for intelligence ( Minsky, 1987). Therefore CycCorp is currently working on steps to dramatically increase the rate at which it can acquire new knowledge. An important piece of this strategy is to integrate *Cyc* with an open source product called *Wordnet* (Fellbaum , 1998).

***WORDNET***

Popular in the computational linguistics community, *WordNet* is a semantic lexicon for the English language. It provides short definitions and synonyms. An easy to use semantic network, *WordNet*'s purpose is twofold: to produce an intuitive combination of dictionary and thesaurus, and to support automatic text analysis. Because *WordNet* has a lexical emphasis and uses a formal taxonomic approach, it is most suitable for lexical categorization and word-similarity determination.

Begun in 1985, *WordNet* was created and is being maintained at the Cognitive Science Laboratory of Princeton University under the direction of Psychology Professor George A. Miller. The database contains roughly 150,000 nouns, verbs and adjectives.

Each *Wordnet* word is organized into 200,000 discrete senses, called *synsets*. Every synset contains a group of synonymous words and *collocations*, which are sequence of words that go together to form a specific meaning, such as "car pool". The meaning of the *synsets* is further clarified with short intuitive definition called a *gloss*. A typical example *synset* with gloss is shown in table 4. Hypernym/hyponym relationships among the noun synsets can be used as an ontology. For example *Wordnet knows* that a *dog* is a *canine*, which is a *carnivore*, which is a *placental mammal*. (See table 5.) Unlike other dictionaries, *WordNet* does not include information about etymology, pronunciation, or the forms of irregular verbs; and contains only limited information about usage.

*Wordnet*'s downloadable knowledge base and browser ran on my XP laptop. Alternatively you can browse the knowledge base on their website. *WordNet*'s architecture reflects the assumption that parts of speech are stored differently in the human brain; thus *Wordnet* relations that connect synsets are different, depending on the part of speech. For example the noun relations are shown in table 3. When a word participates in several synsets, some senses are more common than others. *WordNet* quantifies this by the *frequency score*. *WordNet* also provides *the polysemy count*, which is its number of containing *synsets*.

*WordNet* stores information in flat files called *lexographer files*, which are then processed by a tool called *grind* to produce the database. *Wordnet* infers the root form of a word at run time, and it is not directly stored in the database. Although *grind* and the *lexicographer files* are freely available, modifying and maintaining the database is said to be difficult. The source code is freely available.

| |
|---|
| Y is a hypernym of X if every X is a (kind of) Y |
| Y is a hyponym of X if every Y is a (kind of) X |
| Y is a coordinate term of X if X and Y share a hypernym |
| Y is s a holonym of X if X is a part of Y |
| Y is an eronym of X if Y is a part of X |

**Table 3: *WordNet*'s Relations for Nouns**

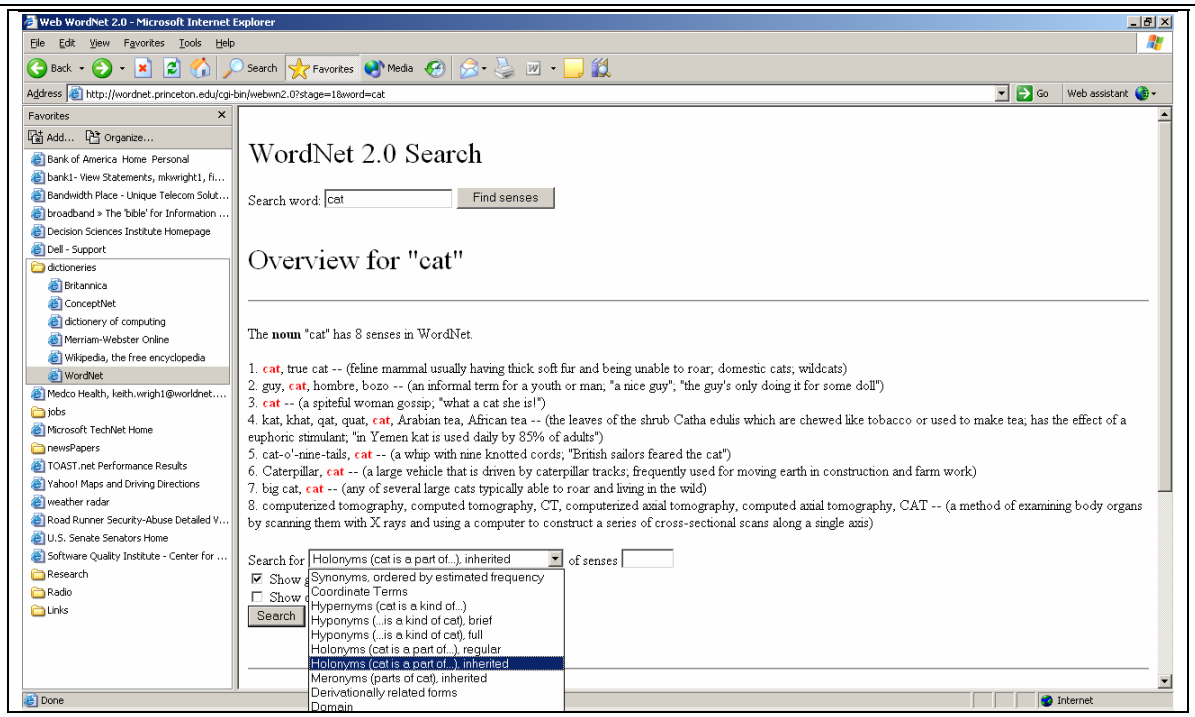


**Table 4: *WordNet* Synset with Gloss**

There are several interesting projects related to *Wordnet*, which are discussed next.

## MISCELLANEOUS COMMON SENSE PROJECTS

One interesting project related to *Wordnet* is the *EuroWordNet* project. It has produced *WordNet*s for several European languages and linked them together, but these are not freely available (http://www.illc.uva.nl/EuroWordNet/).

A related project is the *Global Wordnet* which is coordinating the production and linking of *Wordnets* for all languages (http://www.globalwordnet.org/).

The *eXtended WordNet* project is parsing the *Wordnet* glosses, making them directly computable (http://www.hlt.utdallas.edu/page.php?p=publications). It is also freely available under a license similar to *WordNet*'s.

The *Suggested Upper Merged Ontology* (*SUMO*) is an upper ontology intended as a foundation ontology for a variety of computer information systems. It was developed by *Teknowledg*e Corporation and is one candidate for the *Standard Upper Ontology* that IEEE working group 1600.1 is developing (Pease, A, 2004). *SUMO* was first released in December 2000. It is formulated in a LISP-like language (Pease A, 2004). A mapping from *WordNet* synsets for nouns and verbs to SUMO classes has also been defined. *SUMO* can be downloaded and used freely.

*WordNet* 2.0 Search

Search word: [ ] Find senses

Results for "Holonyms (this is a part of...), inherited" search of noun "cat"

1 of 8 senses of cat

Sense 7
big cat, cat -- (any of several large cats typically able to roar and living in the wild)
MEMBER OF: Felidae, family Felidae -- (cats; wildcats; lions; leopards; cheetahs; saber-toothed tigers)
MEMBER OF: Carnivora, order Carnivora --
(cats; lions; tigers; panthers; dogs; wolves; jackals; bears; raccoons; skunks; and members of the suborder Pinnipedia)
MEMBER OF: Eutheria, subclass Eutheria -- (all mammals except monotremes and marsupials)
MEMBER OF: Mammalia, class Mammalia -- (warm-
blooded vertebrates characterized by mammary glands in the female)
MEMBER OF: Vertebrata, subphylum Vertebrata, Craniata, subphylum Craniata --
(fishes; amphibians; reptiles; birds; mammals)
MEMBER OF: Chordata, phylum Chordata --
(comprises true vertebrates and animals having a notochord)
MEMBER OF: Animalia, kingdom Animalia, animal kingdom --
(taxonomic kingdom comprising all living or extinct animals)

**Table 5: *Wordnet* holynyms for 'cat'**

Next we discuss the *WikiPedia* project, notable because of it's its novel approach of using internet collaboration to address the problem of scale.

## *WIKIPEDIA*

*WikiPedia* is a freely-available multilingual encyclopedia, collaboratively edited and maintained by thousands of volunteer internet users (http://en.wikipedia.org/). The project began in 2001 and is hosted and supported by the non-profit *Wikimedia* Foundation. Each entry in *WikiPedia* is called a *wiki.* In addition to typical encyclopedia entries, *Wikipedia* includes current events and information more often associated with almanacs and magazines. *WikiPedia* currently contains over 400,000 articles in English and over 700,000 in other languages. Any visitor to *Wikipedia* can edit or add to its contents using the *wiki* software.

Since pages are always subject to editing, no article is ever really finished. As such, *Wikipedia* is subject to vandalism; but it has systems in place to deal with these challenges.

In summary, despite the scale of the problems in building intelligent systems, large collaborative projects like *Wikipedia* have rekindled the idea of putting common sense into computers. Researchers at MIT continue to believe that, unless we can first learn how to manually build systems with common sense, we will not be able to build learning machines that can automatically learn common sense (Liu, H. & Singh, P., 2004). In another approach to gathering knowledge from a large number of contributors, they have created *OMCSS*.

## THE OPEN MIND COMMON SENSE SYSTEM (*OMCSS*)

### *Theory*

In the year 2000, inspired by the *Cyc* project and the ideas of Minksy, 1987; Gentner, 1983; and Gelernter, 1994; the MIT media lab created a web site where anyone can contribute pieces of their own common sense (Liu & Singh, 1984). This web site is known as *OMCSS*, the Open Mind Common Sense System (www.openmind.org/commonsense). So far it has gathered over a million pieces of common sense knowledge from over fourteen thousand ordinary lay people. The challenge there is how to capture the knowledge with the minimum error, and how to represent the knowledge in ways to maximize its computability.

The *OMCSS* project is based on the belief that diversity in knowledge representations and inference mechanisms is the secret to success. Therefore, compared to *WordNet*, the knowledge in *ConceptNet* is more informal and more heuristic. *OMCSS* is based in part on the idea that common sense involves the following basic heuristics.

- *If you have a problem, think of a past situation where you solved a similar problem.*
- *If you take an action, anticipate what might happen next.*
- *If you fail at something, imagine how you might have done things differently.*
- *If you observe an event, try to infer what prior event might have caused it.*
- *If someone does something, ask yourself what the person's purpose was in doing that.*

### *OMCSS Knowledge Capture*

To capture knowledge from ordinary web users, *OMCSS* researchers built a variety of fill-in-the-blank web pages. Each page captures a certain kind of knowledge. Some pages ask for descriptions of photos, such as, “A mother is holding her baby. The baby is smiling. They are looking into each other's eyes. The baby is happy. The mother is happy.” On other pages users supply stories, either to illustrate some existing fact like "flashlights light up places". For example, “It was too dark to see. I went and got my flashlight. I turned on my flashlight. I could see much better.” Other pages allow users to annotate movies of simple spatial events. For example, “The small red ball rolls past the big blue ball.” See figure 4 for a complete list of all the different types of knowledge capture pages. The results of some of my own sessions with *OMCSS* are shown in figure 5.

The inference engine and toolkit that the *OMCSS* project uses for inferencing is called *ConceptNet*. It is freely available at http://web.media.mit.edu/~hugo/conceptnet/.

***CONCEPTNET's Capabilities***

According to Liu & Singh, 1984, *ConceptNet*, given a story describing a series of everyday events, can infer where these events will likely take place; the mood of the story; and the possible next events. Furthermore they claim that, given a natural language search query, *ConceptNet* has the potential to determine which meaning is most likely. They also say that when presented with a novel concept appearing in a story, *ConceptNet* can determine which known concepts most closely resemble or approximate the novel concept.

If you go to (http://xnet.media.mit.edu/conceptnet/conceptnet.swf), you can browse the *OMCSS* knowledge network on line. That browser is shown in figure 6, which also shows *OMCSS*'s semantic connections to the word ~~,~~ *wedding*. From this picture we can see the *OMCSS* knows that a wedding may be a relationship between persons; a property of a wedding may be romance; a property of a wedding is that it is sweet; a wedding may be an event; and a wedding is sometimes desired by something.

Alternatively you can download the *ConceptNet* software to your PC. That version of the browser accepts arbitrary text input, but is not graphically oriented (See figure 7.). The downloadable knowledge browser has several functions activated by buttons – *browse*, *guess concept*, etc. From figure 7 we can see that *OMCSS* knows the following additional information about the word wedding: Weddings may happen on *Saturday*. Weddings may be related to the concept of *bride*. Weddings may be related to the concept of *a person's son.* Weddings may be related the concept of confetti, *e*tc.

***CONCEPTNET Architecture***

The *OMCSS* knowledge base is structurally similar to that of *WordNet*, but its scope is as wide as that of *Cyc*. *ConceptNet* increased *WordNet*'s repertoire of three semantic relations -- *synonym*, *is-a*, and *part-of* -- to twenty-three relations including, *EffectOf*, *SubeventOf*, *CapableOf* , *PropertyOf*, *LocationOf*, and *MotivationOf*. (See figure 8.) *ConceptNet*'s reasoning method can best be thought of as graph traversal, where the nodes are simple phrases and the arcs are logic predicates. More than *WordNet* and *Cyc*, *OMCSS* stresses the importance of associational knowledge. Thus, *ConceptNet* invests heavily in making associations between concepts, even ones whose value is not immediately apparent. As a result, about two thirds of *OMCSS's* 118,000 assertions serve only to interrelate other assertions (See figure 8.). These associate predicates are called *kLines* (Minsky, 1987).

The ConceptNet knowledge base is built periodically from the *OMSYS* database, by an automated three-stage process: (1) extraction, (2) normalization , and (3) relaxation.

During *extraction*, each sentence is first parsed by a freely available natural language processor called *MontyLingua*. The fifty extraction rules use regular expressions, syntactic constraints, and semantic constraints to map *OMCSS* structured English sentence fragments to *ConceptNet*'s assertions. Multiple assertions can be inferred from a single *OMCSS* sentence. For example, from the sentence, "A lime is a sour fruit", *ConceptNet* extracts the knowledge, *IsA*(lime, fruit)

and infers *PropertyOf*(lime, sour). During extraction, duplicate assertions are merged, and an additional field called *frequency* is added to each predicate. Sentences for which there are no suitable relation types are put into the generic, *ConceptuallyRelatedTo* relation.

| | | |
|---|---|---|
| Cause and effect | - | Explain the effects of actions |
| Connect the sentences | - | Come up with a connection between two random sentences |
| Describe a picture | - | Describe a picture in a sentence or two |
| Describe a verb | - | Given a verb, what sorts of objects can it apply to? |
| Doing things | - | What actions you take while engaging in an activity |
| Enter a fact | - | Enter sentence with no special prompting |
| Explain a relation | - | Relate a pair of words that are usually near each other |
| Explain why | - | Give a reason why a fact is true |
| First person perspective | - | Describe a situation you might be in |
| Give me a sentence | - | Prompts you with a random sentence |
| Give me a word | - | Prompts you with a random word |
| Goals and desires | - | What sorts of things do people want and not want? |
| Help write a story | - | Help write a story about an everyday activity |
| Illustrate a fact | - | Tell a story that illustrates a fact |
| List objects in scene | - | List the objects you see in a picture |
| List some concepts | - | Give Open Mind some more concepts to think about |
| Paraphrase the sentence | - | Supply another way to say the same thing |
| Respond to a picture | - | Prompts you with a picture to remind you of things you know |
| Sentence patterns | - | Enter sentences that fit a certain pattern |
| Similarities and differences | - | What are the similarities and differences between two things? |
| Spatial concepts | - | Describe in words a spatial idea |
| Tell a ministory | - | Tell a very short story about a single event |
| Uses and functions | - | What things are for and how they are used |
| Uses and functions (many) | - | What things are for, five at a time |
| Using knowledge | - | Given a fact, what is a question it might help answer? |
| What changed? | - | Describe what changed during an event |
| What else should I know? | - | List some other things someone should know to fully understand an event |
| What to do | - | How do you get what you want? |
| Where things are | - | Where are things typically found? |

**Figure 4: Types of Knowledge Capture in the *OMCSS* Web Site**

In the *normalization* phase, errant spelling is corrected, and the text is stripped of determiners like “the”, “a”, and other semantically peripheral words. Then words are stripped of tense and

number. For example *is*, *are*, and *were* become *be*; and *apple* becomes *apple*. In the *relaxation* phase, several things are added to the database to improve performance. First, generalizations are pre-inferred by using the *IsA* relation.

| |
|---|
| Something that might happen while standing up is bumping your he |
| |
| Please connect the following two sentences with as *few intermediate sentences* as possible: Something that might happen as a consequence of fighting the enemy is casualties. Investing money or energy requires having resources.<br>Fighting an enemy requires money<br>Please teach Open Mind how the verb offer can be used. |
| |
| You can offer a service |
| The last thing you do when you prepare for a vote is read |
| |
| Please enter a clear and simple fact about the world.<br>Wheels roll |
| |
| What is the relationship between the word tuna and the word celery?<br>Salad food |

**Figure 5: Interactions with *OMCSS***

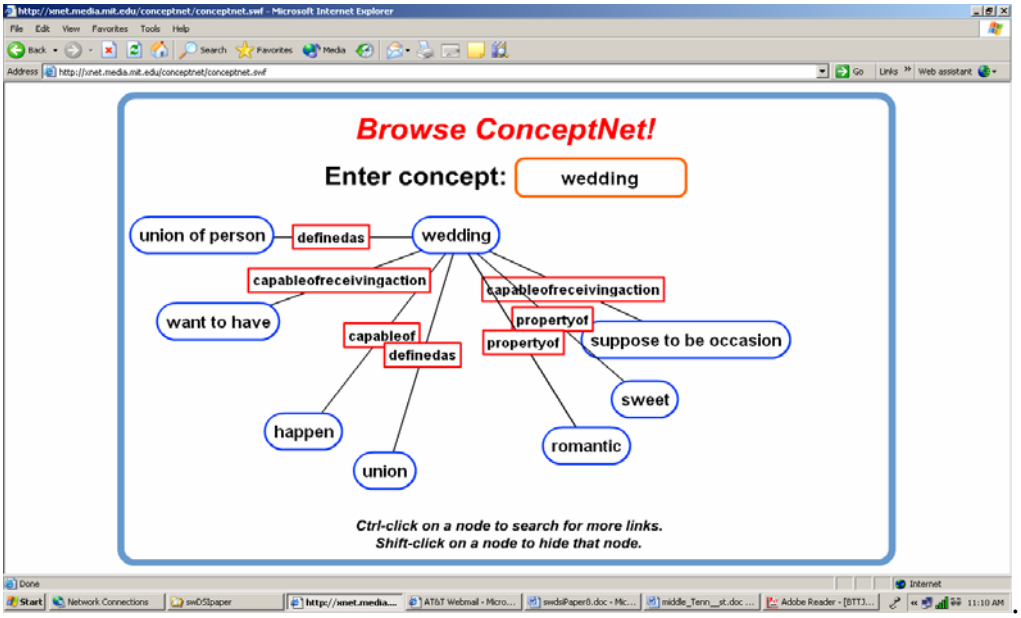


**Figure 6: A session with the *ConceptNet* web site browser**

Next, the *kLine* connection predicates are added (See figure 8.). These dramatically increase the

connectivity of the semantic network, and make it more likely that novel concepts extracted from arbitrary text can be integrated.

There are three *kLine* relations -- (*SuperThematicKLine*, *ThematicKLine*, and *ConceptuallyRelatedTo*).

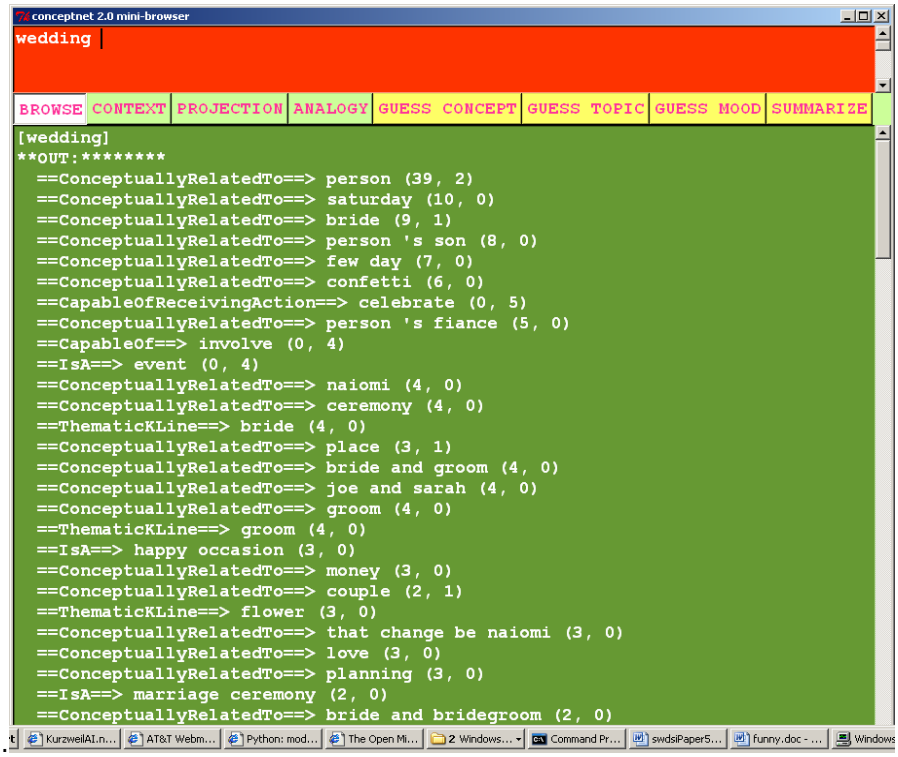


**Figure 7: A session with the downloadable version of *ConceptNet***

*SuperThematicKLines*, unify themes with their variations. For example since “buy” is a super-theme of “buy food”; the predicate to represent this fact would be – (*SuperThematicKLine* ‘buy food’, ‘buy’). This mechanism allows the merging of similar nodes via the use of *WordNet* and *FrameNet* (Fillmore & Lowe,1998). For example, nodes pairs like *bike/ bicycle* and *sad/sadness* are merged.

The backbone *ConceptNet* is a natural (English) language-processing engine called *MontyLingua*, which from raw English sentences, extracts subject/verb/object tuples, adjectives, noun phrases, verb phrases, people's names, places, events, dates, times, and other semantic information. You can customize it by installing your own lexicon. It tokenizes raw English text (sensitive to abbreviations), and resolves contractions, e.g. "you're" ==> "you are". It strips inflectional morphology, i.e. changes verbs to infinitive form and nouns to singular form.
When an English text string is fed into *ConceptNet*, *MontyLingua* extracts the verb-subject-object-object frames. These frames closely resemble the structure needed for the semantic network nodes, and thus inferencing.

The fundamental inference mechanism in *ConceptNet* is called *projection*. Projection is best

described as a graph traversal using a single relation type. Only transitive and ordering assertions can be used in projection. (i.e. *LocationOf, IsA, PartOf, MadeOf, FirstSubeventOf, LastSubeventOf, SubeventOf), EffectOf, DesirousEffectOf).* An example of a spatial projection would be that "Los Angeles is located in California; which is located in United States; which is located on Earth.". For example, researchers could use sub-event projection for goal planning, and causal projection for predicting possible outcomes and next-states. In the *ConceptNe*t browser (figure 7) the *concept guessing* function first uses projection, and then computes the relatedness of connected concepts by the numbers and weights of each arc in the connecting path (Lui & Singh, 2004). Researchers can assign a different set of numeric weights for each problem domain. *OMCSS* researchers are experimenting with what they call *realm-filtering*, which omits certain relation-types; resulting in for example *temporal only* or *spatial only* context neighborhoods. For example, getting only the temporally forward conceptual expansions would be equivalent to imagining possible next states from the current state.

| **K-LINES (1.25 million assertions)** |
|---|
| (ConceptuallyRelatedTo 'bad breath' 'mint' 'f=4;i=0;') |
| (ThematicKLine 'wedding dress' 'veil' 'f=9;i=0;') |
| (SuperThematicKLine 'western civilisation' 'civilisation' 'f=0;i=12;') |
| **THINGS (52 000 assertions)** |
| (IsA 'horse' 'mammal' 'f=17;i=3;') |
| (PropertyOf 'fire' 'dangerous' 'f=17;i=1;') |
| (part of 'butterfly' 'wing' 'f=5;i=1;') |
| (MadeOf 'bacon' 'pig' 'f=3;i=0;') |
| (DefinedAs 'meat' 'flesh of animal' 'f=2;i=1;') |
| **AGENTS (104 000 assertions)** |
| (CapableOf 'dentist' 'pull tooth' 'f=4;i=0;') |
| **EVENTS (38 000 assertions)** |
| (PrerequisiteEventOf 'read letter' 'open envelope' 'f=2;i=0;') |
| (FirstSubeventOf 'start fire' 'light match' 'f=2;i=3;') |
| (SubeventOf 'play sport' 'score goal' 'f=2;i=0;') |
| (LastSubeventOf 'attend classical concert' 'applaud' 'f=2;i=1;') |
| **SPATIAL (36 000 assertions)** |
| (LocationOf 'army' 'in war' 'f=3;i=0;') |
| **CAUSAL (17 000 assertions)** |
| (EffectOf 'view video' 'entertainment' 'f=2;i=0;') |
| (DesirousEffectOf 'sweat' 'take shower' 'f=3;i=1;') |
| **FUNCTIONAL (115 000 assertions)** |
| (UsedFor 'fireplace' 'burn wood' 'f=1;i =2;') |
| (CapableOfReceivingAction 'drink' 'serve' 'f =0;i =14;') |
| **AFFECTIVE (34 000 assertions)** |
| (MotivationOf 'play game' 'compete' 'f =3;i=0;') |
| (DesireOf 'person' 'not be depressed' 'f=2;i=0;') |

**Figure 8*: ConceptNet* Knowledge Predicates**

*ConceptNet*'s analogy feature examines the incoming edges of a node. Two nodes are analogous if they share sets of incoming edges. For example, since *apple* and *cherry* share incoming-edges, [(*PropertyOf* x *red); (PropertyOf* x *sweet*); (*IsA* x *fruit*)], they are analogous. *OMCSS* researchers believe it may also prove useful to apply *realm-filtering* to computing analogies,

preferring to variously emphasize *functional similarity*, *affective similarity, or attribute similarity*.

The *affect sensing* feature attempts to classify the emotional tone of a text into one of six affect categories (happy, sad, angry, fearful, disgusted, and surprised) (Lui & Singh, 2004). Then it assesses the affect of any unclassified concept by finding all the paths which lead to each of these six affectively known categories, and then judging the strength and frequency of each set of paths.

### *Applications Of Collaborative Semantic Networks*

Despite their present day shortcomings, semantic networks like *ConceptNet* may soon be applied in several areas. One such application will be in products for enhancing human creativity. For example, when a novelist depicts a scene, he/she may not be able to visualize all the elements that would later convey a good story about the event. Retrieving knowledge from an *OMCSS*-like knowledge base could give the writer a list of creative ideas to convey in order to build a complete story. A slight variant of this idea could produce a story-generator that allows a person to interactively invent a story with the system using causal projection chains to create storylines.

Another application will be in foreign language phrase translations. For example a smart PDA given a phrase like “I am at a restaurant.”, could automatically generate a list of concepts relevant to the situation like “people”, “waiter”, “chair”, “eat’, and their corresponding translation.

In summary, the theory behind *OMCSS* and *ConceptNet* is intuitive and interesting, and there are a growing number of simple implementations of large collaborative semantic networks. More sophisticated applications will require advances in automated text analysis, of the kind described in Lui, Sing, 2004. Some of the results I had with the freely-available *ConceptNet* text analysis tools were impressive.

### *Using CONCEPTNET For Text Analysis*

Lui &Singh, 2004 claims that when the user types "*my cat is sick*" into the *ConceptNet* browser in the *guess concept* mode, the system reasons, roughly, that – “People care about their pets. People want their pets to be healthy. My cat is my pet. I want my cat to be healthy. A veterinarian heals sick pets.,— A veterinarian makes sick pets healthy. I want to call a veterinarian. A veterinarian is a local service. Therefore: the concept is likely veterinary medicine.” However when I used the downloadable version, I did not see quite that level of reasoning sophistication. Tables 6-12 shows some of my own results. With about a day’s worth of practice I was able to learn what sorts of text entries *ConceptNet* did well with. It seemed to like simple short sentences, each describing some single attribute of the concept I had in mind.

I got the impression that the current knowledge base is much too sparse in most areas. Only the Guess Concept and Guess Topic modes seemed to do anything reasonable, and as you have seen, these were only toy examples.

| My cat is sick. |
|---|
| [is it: fluffy?] (2.0) ==PropertyOf==> sick (2.0) |
| [is it: kirstin?] (2.0)==PropertyOf==> sick (2.0) |
| [is it: woman?] (2.0) ==PropertyOf==> sick (2.0) |
| [is it: unhealthy thing?] (2.0)==PropertyOf==> sick (2.0) |
| [is it: vomitting?] (2.0) ==PropertyOf==> sick (2.0) |
| [is it: have heart attack?] (1.5) ==UsedFor==> be sick (2.0) |
| [is it: disease?] (1.5)==UsedFor==> be sick (2.0) |
| [is it: lose consciousness?] (1.5)==UsedFor==> be sick (2.0) |
| [is it: catch mump?] (1.5)==UsedFor==> be sick (2.0) |

**Table 6: Guess Concept Mode**

| "My car needs a new transmission or it may strand me someplace." |
|---|
| [is it: car?] (3.5) ==PropertyOf==> new (1.0) ==CapableOf==> need (1.0) |
| [is it: baby?] (3.5) ==PropertyOf==> new (1.0) ==CapableOf==> need (1.0) |
| [is it: person?] (2.5) ==CapableOf==> need (1.0) ==DesireOf==> need (1.0) |
| [is it: tissue engineering?] (2.0) ==PropertyOf==> new (1.0) |
| [is it: revelation?] (2.0)==PropertyOf==> new (1.0) |
| [is it: moon?] (2.0) ==PropertyOf==> new (1.0) |
| [is it: day?] (2.0)==PropertyOf==> new (1.0) |

**Table 7: Guess Concept Mode**

| "It has a court. It has a ball. It has a racket. It has a net. It has a serve. It has players." |
|---|
| is it: person?] (2.5) ==CapableOf==> have ball (1.0) ==PartOf==> ball (1.0) |
| [is it: mcdonald?] (1.5) ==CapableOf==> have serve (1.0) |
| [is it: king and queen?] (1.0) ==part of==> court (1.0) |
| [is it: team?] (1.0)==PartOf==> player (1.0) |
| [is it: hear testimony?] (1.0)==part of==> serve (1.0) |
| [is it: jury?] (1.0)==PartOf==> court (1.0) |
| [is it: chess?] (1.0)==PartOf==> player (1.0) |
| [is it: basketball player?] (1.0) ==PartOf==> ball (1.0) |
| [is it: tennis game?] (1.0)==PartOf==> player (1.0) |

**Table 8: Guess Concept mode**

| "The preacher was late. The bride was beautiful. The groom was tall. The cake was white. |
|---|
| groom (39%) |
| bride (33%) |
| preacher (31%) |
| be tall (22%) |
| cake (21%) |
| be late (19%) |
| wedding (15%) |
| white (16%) |
| look (15%) |

**Table 9: Guess Topic Mode**

| |
|---|
| The engine was powerful. It had wheels. It had a trunk.The seats were leather. The windows rolled down. I drove it on the street. People like it. |
| [is it: car?] (9.0)==CapableOf==> drive (2.0) ==LocationOf==> on street (2.0) |
| [is it: person?] (7.0)==CapableOf==> drive (2.0) ==LocationOf==> on street (2.0) |

**Table 10: Guess Concept Mode**

Another possible research area for knowledge bases like *OMCSS* is in *query expansion*. An example of this would be to enter the something like the earlier example into a pre-processor for Google – "List under valued companies on the New York Stock Exchange.". A pre-processor built on a semantic network could potentially translate this into something that a search engine could better understand. Table 11 shows my result with that text in *ConceptNet*.

| |
|---|
| *Give me a list of under-valued companies on the New York Stock Exchange.* |
| is it: smoking?] (1.5)==UsedFor==> value (1.0) |
| [is it: miser?] (1.5)==CapableOf==> value (1.0) |
| [is it: moron?] (1.5) ==CapableOf==> value (1.0) |
| [is it: appraiser?] (1.5)==CapableOf==> value (1.0) |
| [is it: poor person?] (1.5)==CapableOf==> value (1.0) |
| [is it: person?] (1.0)==DesireOf==> value (1.0) |

**Table 11**

As you can see, *ConceptNet* was unable to make sense out of that text.

But when I remembered what *ConceptNet* likes, which are a few short simple sentences, I found that *ConceptNet* did have some knowledge of businesses.

| |
|---|
| It had earnings. It showed profit. It had a balance sheet. It had stock. It had employees. |
| [is it: bookstore?] (1.0) ==part of==> employee (1.0) |
| [is it: company?] (1.0) ==part of==> earnings (1.0) |

**Table 12**

## CONCLUSIONS AND FUTURE RESEARCH

In conclusion, there are many new and interesting freely available software tools and documentation that purport to aid research in building intelligent text mining systems, or those that can exhibit some form of common sense. The projects that provide these tools are most notably -- *Cyc*, *WordNet*, and *OMCSS*. Whereas *WordNet* can compute word-similarity, and *Cyc* can do formalized logical reasoning, *ConceptNet* can do contextual inferencing. Of these, only *ConceptNet and WordNet* are open source. However the others offer full inspection of their respective knowledge bases. This paper reviewed my experiences with some of these tools.

The *openCyc* tools come with a wealth of very interesting documentation and tutorials; but the tools seemed complex, unstable, and unlikely to be useful for independent research. Such research projects would likely require more support from openCyc.org than is now freely available. But the biggest argument against doing research on formal common sense reasoning like that of *Cyc* is that it has proven prohibitively time consuming and expensive.

On the other hand, collaborative semantic network projects like *OMCSS* are relatively inexpensive. Furthermore, the *ConceptNet* tools were easy to understand, and lived up to many of the claims made by their creators (Lui & Singh, 2004). Furthermore, since *ConceptNet* provides source code, it would be an interesting future research project to closely examine that source, trace exactly how it works, and build an application on top of it. *ConceptNet 2.1* is distributed both as a *Python* language API, and a standalone XML-RPC Server. Thus researchers will need to also download and install the freely available *Python* interpreter. If you want to access *ConceptNet* from other programming languages such as Java or C++, you'd launch *ConceptNet*'s XML-RPC Server and then interface with *ConceptNet* via an XML-RPC client, which is available for all major programming languages. Sample client code is available. There are no supposedly no differences between the free *ConceptNet* tools and the ones used at the MIT media lab (Singh, P, 2004).

The obvious disadvantage to these types of collaborative semantic network projects is the potentially poor quality of the knowledge, which is entered by lay people; some of whom are vandals. Another consideration is the probable lack of freely available tool support. I was able to get some assistance from the MIT media lab, but not very much (Sing., P. 2004).

A truly commonsense reasoning system will need to operate on many levels. Research over the last twenty years has shown that no single type of inference strategy, is by itself up to the task of commonsense reasoning. This is still a poorly understood area. A basic question is how to manage multiple types of thinking processes?

Future research on semantic networks like *Concept Net* will likely focus on the network enhancement or *relaxation* phase, where there is much potential for improvement. A likely result will be tools to correct the system when it inferences incorrectly, or fails to make a possible inference.

Future research on formal reasoning system like *Cyc* will focus on tools to automate the knowledge acquisition process which has thus far proved too expensive.

I believe that common sense databases will some day be ubiquitous. One way or the other, through directly acquiring knowledge from people, or reading the web, we will find ways to encode a vast amount common sense. But such knowledge bases by themselves are not nearly enough to build systems with human-level thinking abilities.

## REFERENCES

Baker C F, Fillmore C J and Lowe J B: 'The Berkeley FrameNet project', in Proceedings of the COLING-ACL, Montreal, Canada, 1998.

Collins A and Loftus E: ‘A spreading-activation theory of semantic processing’, Psychological Review, 82, No 6, pp 407—428 (1975).

Davies, J, Fensel, D, and Harmelen, F editors. ‘Towards the Semantic Web: Ontology driven Knowledge Management’. John Wiley and Sons, Ltd., 2003.

De Oliveira, J. CycCorp Marketing Manager. Interview, November 2004.

Fellbaum C (Ed): ‘WordNet: An Electronic Lexical Database’, MIT Press, 1998.

Gelernter D: ‘The Muse in the Machine: Computerizing the Poetry of Human Thought’, Free Press, 1994.

Gentner D: ‘Structure-mapping: A theoretical framework for analogy’, Cognitive Science, 7, pp 155—170, 1983.

Lenat D B: ‘CYC: A large-scale investment in knowledge infrastructure’, Communications of the ACM, 38, No 11, pp 33— 38 , 1995.

Liu, H. & Singh, P. ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal, To Appear. Volume 22, forthcoming issue. Kluwer Academic Publishers, 2004.

Minsky M: ‘The society of mind’, Simon & Schuster, 1987.

Pease, A. “Standard Upper Ontology Knowledge Interchange Format” http://cvs.sourceforge.net/viewcvs.py/*checkout*/sigmakee/sigma/suo-kif.pdf (http://home.earthlink.net/~adampease/professional/), 2004.

Reed, Stephen, & Lenat. (2002) 'Mapping Ontologies into Cyc', AAAI Conference Workshop on Ontologies For The Semantic Web, July 2002, Edmonton, Canada, 2002.

Singh, P. email, December 2004.