Open Mind Common Sense: Knowledge Acquisition from the General Public

Push Singh[1], Thomas Lin[1], Erik T. Mueller[2],
Grace Lim[1], Travell Perkins[1], Wan Li Zhu[1]

[1] MIT Media Laboratory, 20 Ames Street, Cambridge, MA 02139 USA
{push, glim, tlin, markt, wlz}@mit.edu,
travell_perkins@yahoo.com
[2] IBM Thomas J. Watson Research Center, P.O. Box 704,
Yorktown Heights, NY 10598, USA
etm@us.ibm.com

Abstract. Open Mind Common Sense is a knowledge acquisition system designed to acquire commonsense knowledge from the general public over the web. We describe and evaluate our first fielded system, which enabled the construction of a 450,000 assertion commonsense knowledge base. We then discuss how our second-generation system addresses weaknesses discovered in the first. The new system acquires facts, descriptions, and stories by allowing participants to construct and fill in natural language templates. It employs word-sense disambiguation and methods of clarifying entered knowledge, analogical inference to provide feedback, and allows participants to validate knowledge and in turn each other.

1 Introduction

We would like to build software agents that can engage in commonsense reasoning about ordinary human affairs. Examples of such commonsense-enabled agents are:

- SensiCal, which reminds a user not to take a vegetarian friend to a steakhouse [1],
- the Cyc team's image retrieval program, which retrieves a photo of a grandmother with her grandchild given the query "happy person" [2], and
- REFORMULATOR, which searches for local veterinarians when a user enters "my cat is sick" [3].

Few such common-sense agents currently exist, and those that do have only been demonstrated to work on select examples to demonstrate the promise of applying common sense. However, as perceptive environments emerge and software becomes more context-aware, the need for reasoning about the ordinary human life will only increase.

What is holding back the development of such applications? While there has been much work on developing representations and reasoning methods for commonsense domains [4], and on the logical underpinnings for commonsense reasoning [5], there has been far less work on finding ways to accumulate the knowledge to do so in practice. The most well-known attempt has been the Cyc project [6] which contains 1.5 million assertions built over 15 years at the cost of several tens of millions of dollars. Knowledge bases this large require a tremendous effort to engineer. With the exception of Cyc, this problem of scale has made efforts to study and build commonsense knowledge bases nearly non-existent within the artificial intelligence community.

2 Turning to the general public

In this paper we explore a possible solution to this problem of scale, based on one critical observation: *Every ordinary person has the common sense we want to give our machines.* The advent of the web has made it possible for the first time for thousands of people to collaborate to construct systems that no single individual or team could build. Projects based on this idea have come to be known as *distributed human projects*. An early and very successful example was the Open Directory Project, a Yahoo-like directory of several million web sites built by tens of thousands of topic editors distributed across the web. The very difficult problem of organizing and categorizing the web was effectively solved by distributing the work across thousands of volunteers across the Internet.

It is now possible for smaller groups within the artificial intelligence community to build systems that require large amounts of knowledge by engaging the general public. What are the issues that are raised when knowledge acquisition systems turn to the general public, employing thousands of people instead of just a few? The Open Mind Initiative [7] was formed with the goal of studying this issue and applying these kinds of distributed approaches to the problems faced by AI researchers. As part of this initiative, we began the *Open Mind Common Sense* project. The goal was to study whether a relatively small investment in a good collaborative tool for knowledge acquisition could support the distributed construction of a commonsense database by many people in their free time. In this paper we report on our progress so far.

This paper is organized as follows. In the first section we review the first version of the Open Mind Common Sense knowledge acquisition system, present an evaluation of the database accumulated by the system, and describe some applications that have been built using this database. In the second section we present the more sophisticated second-generation version of the system under development and soon to be deployed. The new system uses strict natural language templates, lets participants design those templates, employs word-sense disambiguation and methods of clarifying entered knowledge, engages in analogical inference to provide feedback, and allows participants to validate knowledge and in turn each other.

3 Open Mind Common Sense

The original Open Mind Common Sense system[1] (OMCS-1) is a commonsense knowledge acquisition system targeted at the general public. It is a web site that gathers facts, rules, stories, and descriptions using a variety of simple elicitation activities [3]. Some of the items collected include:

- *Every person is younger than the person's mother*
- *A butcher is unlikely to be a vegetarian*
- *People do not like being repeatedly interrupted*
- *If you hold a knife by its blade then it may cut you*
- *If you drop paper into a flame then it will burn*
- *People pay taxi drivers to drive them places*
- *People generally sleep at night*

OMCS-1 has been running on the web since September 2000. As of August 2002 we have gathered 456,195 pieces of commonsense knowledge from 9296 people. Thousands of people, many with no special training in computer science or artificial intelligence, have participated in building the bulk of the database.

There have been a number of efforts in recent years to develop knowledge acquisition systems that can acquire knowledge from people with no formal training in computer science [8]. One of the key problems such systems address is that end users do not know formal languages. We had considered using the Cyc representation for our project, but it was clear that few members of the general public would be willing to spend the time to learn CycL or the thousands of terms in the Cyc ontology. Some approaches deal with the problem by finding a way to present a constrained natural language interface to the system. One method is to use pull down menus from which the user can select English forms consistent with the underlying representation [9]. Another method is to develop a subset of English which is restricted enough to be easily parsed into first-order logic [10].

We were concerned with overly restricting our users by imposing our own ontological preconceptions, so we took a different approach, which was to allow users to supply knowledge in free-form natural language. We constructed a variety of activities for eliciting this knowledge. One activity was to present the user with a simple story and ask for knowledge that would be helpful in understanding that story. For example, given the story "*Bob had a cold. Bob went to the doctor.",* the user might enter the following:

- *Bob was feeling sick*
- *Bob wanted to feel better*
- *The doctor made Bob feel better*
- *People with colds sneeze*
- *The doctor wore a stethoscope around his neck*
- *A stethoscope is a piece of medical equipment*
- *The doctor might have worn a white coat*

---

[1] http://www.openmind.org/commonsense

- *A doctor is a highly trained professional*
- *You can help a sick person with medicine*
- *A sneezing person is probably sick*

In choosing to acquire knowledge in free-form natural language, we shifted the burden from the knowledge acquisition system to the methods for using the acquired knowledge. We have taken two approaches:

1. Use the English items directly for reasoning. We describe some experiments in reasoning with English syntactic structures in [3]. A few other systems have used natural-language-like representations as an underlying representation, such as the Pathfinder causal reasoning system [11].

2. Use information extraction techniques to convert English items into more standard knowledge representations. There has been significant progress in the area of information extraction from text [12] in recent years, due to improvements in syntactic parsing and part-of-speech tagging. A number of systems are able successfully to extract facts, conceptual relations, and even complex events from text.

This latter approach is a very different way to think about how to go about building a commonsense database. Rather than directly engineering the knowledge structures used by the reasoning system, we instead encourage people to provide information clearly in natural language and then extract from that more usable. As a knowledge acquisition method it is closer in spirit to approaches that apply learning and induction techniques to learn rules from examples supplied by users [13].

We have developed extraction patterns to mine hundreds of types of knowledge out of the database into simple frame representations. Some examples include:

[a | an | the] N1 (is | are) [a | an | the] [A1] N2
→ *Dogs are mammals*
Hurricanes are powerful storms

a person [does not] want[s] to V1 A1
→ *A person wants to be warm*
A person wants to be attractive

N1 requires [a | an] [A1] N2
→ *Writing requires a pen*
Bathing requires water

4 Evaluating the accumulated database

Evaluation is difficult but important for any knowledge acquisition effort. A manual evaluation was performed on the OMCS-1 database to assess its quality and composition. 3245 unique items were collected (about 1% of the database of 432,552 items). Of these, 236 (7.3%) nonstandard items were automatically discarded. Nonstandalone items are those requiring additional materials such as images and stories in order to make sense. The remaining 3009 items were distributed among 7 judges. Of these, 370 (12.3%) were marked by the judges as being garbage. Examples of garbage were:

- *it has a meaning*
- *gone to lunch*
- *you are*

The remaining 2639 items were rated on a scale from 1 to 5 for the following attributes: generality (1=specific fact, 5=general truth), truth (1=false, 5=true), neutrality (1=biased, 5=neutral), and sense (1=makes no sense, 5=makes complete sense).

Results are shown in Fig. 1 (NA=no answer). The average rating for generality was 3.26, reflecting the fact that items of common sense may range from the specific to the general. Sample items rated 5 for generality were:

- *Birds often make nests out of grass.*
- *Dew is wet*
- *Round objects roll with greater ease than other shapes*

Sample items rated 1 for generality were:

- *Eritrea is part of Africa*
- *Tom Smothers knows how to play with yo-yo's*

The average rating for truth was 4.28, with 75% of items rated 4 and higher. 67% of items were rated 5, reflecting the presence of exceptions in many statements of common sense. Sample items rated 5 for truth were:

- *An outfit is something that might have buttons.*
- *houses have many kinds of roofs*
- *Legal matters can be confusing to most humans.*

Sample items rated 4 for truth were:

- *a person wants to be successful.*
- *Small cars are uncomfortable*

Sample items rated 1 for truth are:

- *someone can be at infinity*
- *time flys like an arrow; fruit flies like a banana*

The average rating for neutrality was 4.42, with 82% of items rated 4 and higher, indicating that the database is judged to be relatively unbiased. Sample items rated 1 for neutrality are:

- *Idiots are obsessed with star trek.*
- *Men should do the laundry*

The average rating for sense was 4.55, with 85% of items rated 4 and higher. Sample items rated 1 for sense are:

- *There are limits to how English words may be spelled.*
- *cows can low quietly*

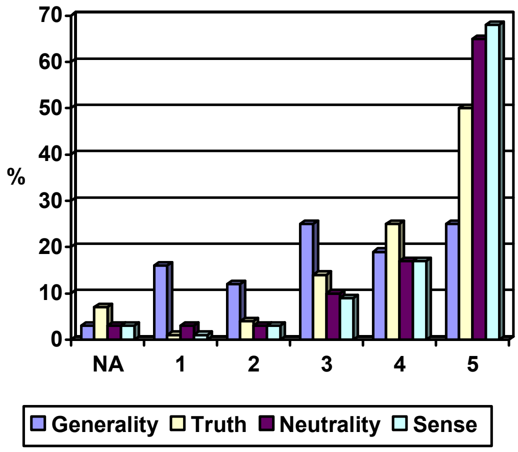

Fig. 1. Manual evaluation of database sample

Judges were also asked to rate sentences for age level. Results are shown in Fig. 2. Most (84%) items were at the grade school or high school level, indicating that the database consists mostly of items that most people know.

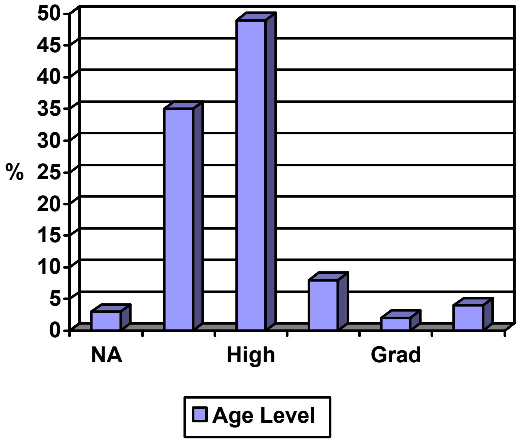

Fig. 2. Contributions by age level

In the next version the users themselves judge the quality of the data, which, of course, requires that they also judge each other.

5 Commonsense-based applications

The database we have accumulated has proven useful in prototyping a number of commonsense-based applications. Our database has been used to build search engines that can reason about users' goals. Novice search engine users naturally express search goals rather than topic keywords. Both REFORMULATOR [3] and GOOSE (goal-oriented search engine) [14] use common sense to infer the query that can most effectively satisfy the search goal. For example, when the user enters "my cat is sick", the system makes the inference that because people care about their pets, to care about something means you want it to be in good health, and that veterinarians can heal sick animals, that the search engine should search for a veterinarian.

A second application that has been developed is the ARIA program, which manages people's personal photos [15]. ARIA uses a spreading activation network of 50,000 sentences from OMCS-1 –statements about object classifications, spatial relations, object purposes, causal relations between events, and emotions resulting from experiencing objects and events –to improve the retrieval of annotated photos. For example, given the information "Susan is Jane's sister", the system can use the fact "in a wedding, the bridesmaid is often the sister of the bride" to retrieve a photo annotated with the text "Susan and her bridesmaids".

A third application is being developed to provide feedback to documentary videographers during shooting process. In the Cinematic Commonsense project [16] OMCS-1 relations and scripts relevant to the documentary subject domain are retrieved to assist the filmmaker in filming content for a documentary subject and recognizing story threads that emerge as content is gathered. After a shot is recorded, metadata is created by the videographer in natural language and submitted as a query to the OMCS-1 database. For example, the shot metadata "a street artist is painting a painting" would yield the shot suggestions such as "the last thing you do when you paint a painting is clean the brushes" and "red paint is expensive." Knowledge about the order of typical events in the painting domain can also be retrieved to create a framework for a sequence of shots.

6 Open Mind Common Sense 2

We have found enough applications and interest that we have begun work on a next-generation version of the system, OMCS-2. While this version has not yet been launched to the general public, we report here on our results so far. The design of OMCS-2 was driven by the lessons learned from the first system, and we will introduce each feature as a way to correct a deficiency in the first system. We learned the following from our experience with OMCS-1:

1. Different participants prefer to enter different types of knowledge.
2. The template activities were the most efficient and most usable form of knowledge.

3. The participants wanted the interaction to be more engaging and provide a sense of utility.
4. The participants wished they could assess, clarify and repair the knowledge.
5. Participants wished they could do more to organize the entered knowledge.

7 Workflow model for acquisition

A major difference between acquiring knowledge from the general public and acquiring it from experts or end users is that the general public is likely to leave as soon as they encounter something difficult. But this does not mean that there should be nothing painful or tedious in the system. Different people like to do different things. Some like to enter new items. Others like to evaluate items. Others like to refine items.

Our system is therefore based on a *distributed workflow* model where the different stages of knowledge acquisition, as in the elicitation, refinement, and restructuring stages of [13], may be performed separately by different participants. The output of the workflow is a *finalized* piece of knowledge, one that has its word senses tagged, clarified, validated, and can participate in some inference. This also gives the participants greater control over their experience.

OMCS-2 allows both template-based input and free-form input. One workflow sequence for template-based entry is as follows:

1. The user browses items in the database until finding an item associated with a template the user is interested in. This frees the user from having to learn the ontology of templates. Instead, a template is located by example.
2. The user then clicks on the template and is given a new input form for that template, along with example items based on that template.
3. After the user enters an item, inferences that result from the item are presented.
4. The user is then given the option of accepting or rejecting those inferences. The accepted and rejected inferences, along with the rules that produced the inferences, are all tagged as accepted or rejected and added to the database. Thus a single interaction supplies many different types of knowledge.

Entered items are also spell checked, tagged for part of speech, and disambiguated as to word sense.

The user may also enter a free-form sentence. The user is informed if the sentence matches an existing template. A template editor may also be brought up, enabling the user to create a new template through variabilization of the entered sentence.

When the user clicks on an item, the user is presented with various activities for criticizing and refining that item.

We now discuss aspects of OMCS-2 in more detail.

8 Templates for knowledge entry

The most useful items gathered by OMCS-1 were those for which we could write information extraction procedures. OMCS-2 therefore encourages knowledge to be supplied using templates rather than as free-form English text. We did not want to use templates in OMCS-1 because we were worried that we would not ourselves be able to design a sufficiently large and fully encompassing set of templates, and we wanted to learn an ontology of relations from our users, instead of imposing one upon them. But since we have now collected several hundred thousand free-form facts from which we can extract templates, we no longer feel we are imposing an ontology on our users. Further, we allow our users to extend the template library themselves.

These templates can extend across multiple lines, so the user can enter descriptions and simple stories extending across multiple sentences. From these we hope to extract larger causal and temporal constraints between states and events, which lets us build structures such as frames and scripts, which we believe are critical for commonsense reasoning. This type of knowledge has not been extensively accumulated in previous commonsense acquisition efforts. Mueller compared several systems and found that most systems were acquiring facts and rules, and not cases and stories against which analogical reasoning could be performed [17]. Examples of these templates include:

?N1 is ?ADJ
?N1 ?V [a] ?N2
?N1 is not ?ADJ
→ *Bob is hungry*
Bob eats a sandwich
Bob is not hungry

The initial set of OMCS-2 templates is based on the templates we extracted from the OMCS-1 database.

9 Feedback through inference

OMCS-1 participants complained that there was no interesting feedback upon entering an item. They wanted some evidence the system could use the item, to feel that they were contributing to the construction of a "thinking machine" and not just a static database. In OMCS-2 we have incorporated several inference mechanisms into the acquisition cycle. The system induces inference rules from the knowledge that people have supplied, and these rules are used immediately to feed back inferences on entered items.

OMCS-2 engages in three types of inference. These methods are simple and fast, and while they do not necessarily produce accurate rules, this is less of a problem since the user is in the loop. The inference depends on the knowledge in the database being stored as instances of templates. The database is a graph of concepts and n-ary relations indexed for retrieval by concept or relation or both. The graph is much larger version of that shown in Fig. 3.

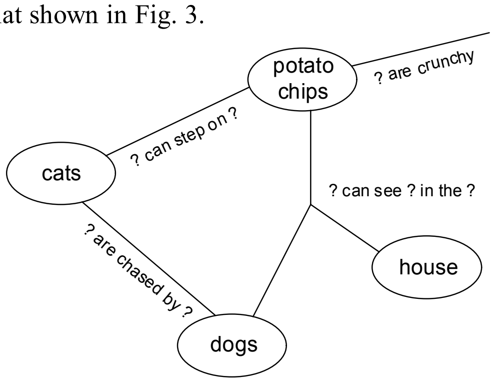

Fig. 3. In this graph, nodes are the "concepts" and links are the "relations." Notice that a relation can connect any number of concepts.

9.1 Method 1: Analogies over Concepts

The first method finds analogies over concepts:

1. A user enters "A mother can have a baby" as an instance of the template "A ? can have a ?" with the concepts "mother" and "baby."
2. The program finds all other items in the database relating "mother" and "baby," such as "A mother can hold her baby."
3. Each such item is an instance of a template. In this case, "A mother can hold her baby" is an instance of the template "A ? can hold her ?" with the concepts "mother" and "baby."
4. Then it finds all other instances of this template, such as "A small girl can hold her small dog."
5. For each one, it instantiates the original template with the concepts. In our example, it yields "A small girl can have a small dog," which is fed back to the user.

9.2 Method 2: Analogies over Relations

The second method finds analogies over relations:

1. A user enters "A mother can have a baby" as an instance of the template "A ? can have a ?" with the concepts "mother" and "baby."
2. The program finds all other sets of concepts involved in instances of this template, such as "A child" and "goldfish" in "A child can have a goldfish."
3. For each such set of concepts, the program finds other instances of templates involving the concepts. For example, "child" and "goldfish" are also involved in

"A child can take care of a goldfish," an instance of the template "A ? can take care of a ?."
4. Each of these templates is then instantiated with the original concepts. Here, we get "A mother can take care of a baby," which is fed back to the user.

Using this system, the program has used "Hawks eat rabbits" to infer "There is more rabbits than there are hawks" by relating (hawks, rabbits) to (cows, grass) and finding "There is more grass than there are cows."

9.3 Method 3: Analogies as Inference Rules

The third method performs analogical inference by first generating a list of inference rules. This is achieved by identifying cycles in the graph in which 3 binary relations connect three concepts, such as that shown in Fig. 4.

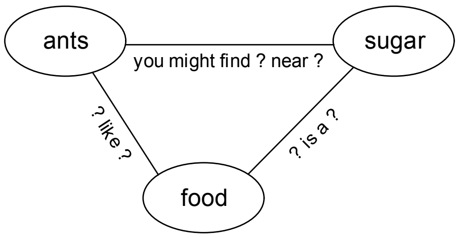

Fig. 4. In this graph, "ants," "sugar," and "food" are in a cycle. The three original sentences are: "You might find ants near sugar," "Ants like food," and "Sugar is a food."

Whenever the program finds such a cycle, it produces a new inference rule. When run on the original OMCS-1 database, 1,860,182 such inference rules are automatically extracted by this method. Inference rules are currently judged as "better" if they match more occurrences in the original database. For each such rule, the program can identify other places in the graph where any two of the three elements in the inference rule can be instantiated. The program then pieces together what a third element must look like to be consistent with the first two, and presents it as an inference.

If the user enters "Bats like darkness" and "You might find bats near cave interiors" is already in the database, then the program matches: ?a = "bats," ?b = "darkness," and ?c = "cave interiors" to infer "Cave interiors is a darkness." While this new sentence is not syntactically correct, it expresses the new idea that "Cave interiors are dark" which was not originally in the database.

Using this system, the program has used "The piano is in the lake" to infer "If you want to go to the piano, then you should take a boat." It has also used "The Christmas Tree has candy canes" to infer "The Christmas Tree is covered mostly by sugar."

These are of course the examples where the method succeeds.

We are presently extending these techniques to allow the system to hypothesize not just inference rules, but also chains of stories. This allows for some limited forms of temporal reasoning by finding analogies between stories. Narratives and stories are unified into a single template representation in which either facts, descriptions, or stories may be expressed.

10 Clarification

It is important to develop techniques to simplify and disambiguate the contributed knowledge.

10.1 Restricting vocabulary

OMCS-1 participants often entered expert rather than commonsense knowledge. One way to make it more likely for participants to enter commonsense knowledge is to encourage the use of common English words. We formed a set of common words by ranking words in OMCS-1 by number of occurrences. In OMCS-2, we encourage the use of common words by suggesting replacements for uncommon words used in entered items. Replacements are suggested through the use of a synonym and hypernym dictionary such as WordNet. The user may accept or reject the replacements.

10.2 Word sense disambiguation

OMCS-1 participants were asked to enter items in "simple English." Though this produced items easy to parse syntactically, it also resulted in items that were difficult to disambiguate since common words are the most polysemous! Word-sense disambiguation is a well-known problem in natural language processing, and a variety of automated and semi-automated methods exist for dealing with the problem. To date the best automated methods achieve about 70% accuracy [18].

The Open Mind Word Expert web site [19] was recently launched as an experiment in gathering word sense information from the general public. Similarly, OMCS-2 introduces word-sense disambiguation into the workflow. Participants are not required to tag the sense of every word in every sentence. Rather, automated methods are used to suggest sense tags, which may be corrected by the user. Further, given that the user has disambiguated some of the words, the system should be able to disambiguate the rest of the words automatically. The user should only have to disambiguate one or two words out of every sentence to pin down the meanings of the other words.

11 Organization

OMCS-1 participants wished they could organize the database to make browsing easier, and application developers using the database wished they could quickly acquire knowledge on particular topics.

We ask users to supply not only knowledge items, but ways to index and organize those items to facilitate retrieval and application. To do this we have our users build *topic vectors*, sets of concepts that are related to a given topic. These are initially built automatically by looking at the words that are correlated with the topic word. Users can then increase and decrease the probability of membership of individual concepts, and add new concepts as well. Topic vectors are commonly used in knowledge retrieval to cast a wider net in order to retrieve all knowledge relevant to a topic. These might also be used as a dynamic way of generating reasoning contexts. In Cyc every assertion belongs to a fixed microtheory and this requires the user to know all the microtheories. In our system the user only has to build topic vectors, and never has to manually arrange the knowledge into particular microtheories.

12 Validation and repair

In a large collaborative effort it is important to assess user honesty. Our present set of participants is relatively small and so it has been fairly easy to filter troublesome users manually. But if the system grew by an order of magnitude, manually filtering users would become too time-consuming. OMCS-2 incorporates mechanisms for peer review, enabling users to judge each other by judging samples of each other's knowledge. By giving users "trust" ratings, the judgments of users with higher trust ratings are given greater weight. We also employ catch trials, as described in [20], in which a fixed set of pre-validated sentences are presented to users from time to time in order to assess their truthfulness.

In order to suggest items for review, we use plausibility measures. If words appear together in ways that are implausible statistically, the system raises an alarm and posts that item for review.

OMCS-2 enables reviewed items to be corrected, subject to further review.

13 Conclusion

We have built the second largest database of commonsense knowledge, after Cyc. In this paper we presented Open Mind Common Sense, a system for acquiring commonsense knowledge from the general public. We described our experiences with the first system OMCS-1 and how they motivated the design of OMCS-2. We presented a manual evaluation of the quality of the database produced by OMCS-1 and discussed several prototype applications built with the database using various inferencing techniques.

What issues arise when knowledge acquisition systems turn to the general public, using thousands of people instead of, as is typical, just a few? Participants should be able to enter many forms of knowledge for commonsense reasoning. Participants should be able to enter knowledge using a friendly interface that seems invisible. Individuals like to do certain things, and they will not necessarily be careful about all aspects of the knowledge they enter. Therefore participants should be able to organize and repair each other's pieces of knowledge, and validate each other. Participants should be able to teach more intricate things such as inference rules by example. Participants should feel after entering an item that the system can use it. Motivation is critical.

What future work is suggested by our approach? We wish to absorb more of the powerful ideas that have been developed by the knowledge acquisition community such as methods for acquiring procedural knowledge by example [21], using knowledge acquisition scripts for coordinating changes to the database [22], and turning to more reflective architectures [23] that can understand and detect problems in the knowledge that users have put in, and pose them back to other users for clarification and repair. We wish to allow artificial intelligence researchers to map templates onto logical formulas and to existing ontologies. We wish to give participants a greater degree of control regarding inferencing and procedural knowledge. Our approach has yet to deal with many of the hardest issues in commonsense reasoning such as contexts, exceptions, combining narratives, elaboration tolerance, and others.

Open Mind Common Sense is the first attempt at realizing the idea that we might distribute the problem of constructing a system with common sense. We are excited because we believe that work on building commonsense databases is no longer only the domain of multi-million-dollar "Manhattan projects", and can now be pursued by the distributed artificial intelligence community as a whole and by turning to the general public to achieve what is too difficult and expensive to be achieved by any one group. There is a goldmine of opportunity for people who are willing to accept that there are countless people out there who would be willing to participate as volunteers in the effort to help artificial intelligence researchers build databases larger than any one group could build.

Acknowledgements

We extend our thanks to David Stork for organizing the larger Open Mind effort, and the many thousands of members of the general public who contributed their knowledge to our database. This project was supported by the sponsors of the MIT Media Lab.

References

1. Mueller, E. T. 2000. A calendar with common sense. In Proceedings of the 2000 International Conference on Intelligent User Interfaces, 198-201. New York: Association for Computing Machinery.
2. Guha, R.V. and Lenat, D. B. 1994. CYC: Enabling agents to work together. Communications of the ACM 37(7): 127-142.
3. Singh, P. 2002. The public acquisition of commonsense knowledge. In Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access. Palo Alto, CA, AAAI.
4. Davis, E. 1990. Representations of commonsense knowledge. San Mateo, Calif.: Morgan Kaufmann.
5. Shanahan, M. 1997. Solving the frame problem. Cambridge, Mass.: MIT Press.
6. Lenat, D.B. 1995. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11): 33-38.
7. Stork, D. 1999. The OpenMind Initiative. IEEE Intelligent Systems 14(3):19-20.
8. Blythe, J.; Kim, J.; Ramachandran, S.; and Gil, Y. 2001. An integrated environment for knowledge acquisition. In Proceedings of the 2001 International Conference on Intelligent User Interfaces, 13-20.
9. Blythe, J. and Ramachandran, S. 1999. Knowledge acquisition using an English-based method editor. In Proceedings of the Twelfth Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Alberta.
10. Fuchs, N. E. and Schwitter, R. 1996. Attempto Controlled English (ACE). CLAW 96, First International Workshop on Controlled Language Applications, University of Leuven, Belgium.
11. Borchardt, G. C. 1992. Understanding Causal Descriptions of Physical Systems. In Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, CA, 2-8.
12. Cardie, C. 1997. Empirical Methods in Information Extraction, AI Magazine, 65-79.
13. Bareiss, R.; Porter, B.; and Murray, K. 1989. Supporting start-to-finish development of knowledge bases. Machine Learning 4, 259-283.
14. Liu, H., Lieberman, H., Selker, T. 2002. GOOSE: A Goal-Oriented Search Engine With Commonsense. Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, (AH2002) Malaga, Spain.
15. Lieberman, H. and Liu. H. 2002. Adaptive Linking between Text and Photos Using Common Sense Reasoning. In Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, (AH2002) Malaga, Spain.
16. Barry, B. & Davenport. G. 2002. *Why Common Sense for Video Production?* (Interactive Cinema Technical Report #02-01). Media Lab, MIT.
17. Mueller, E. T. 1999. A database and lexicon of scripts for ThoughtTreasure. CogPrints ID cog00000555 http://cogprints.soton.ac.uk/
18. Ide, N. and Véronis, J. (Eds.) 1998. Special Issue on Word Sense Disambiguation. Computational Linguistics, 24(1).
19. Chklovski, T. and Mihalcea, R. 2002. Building a Sense Tagged Corpus with Open Mind Word Expert. In Proceedings of the Workshop on "Word Sense Disambiguation: Recent Successes and Future Directions", ACL 2002.
20. Stork, D. 2001. Toward a Computational Theory of Data Acquisition and Truthing. In Proceedings of Computational Learning Theory (COLT 01), David Helmbold (editor), Springer Series in Computer Science, 2001.
21. Cypher, A. 1993. Bringing programming to end users. In Watch What I Do: Programming by Demonstration. Cambridge, Mass: MIT Press.

22. Tallis, M. and Gil., Y. 1999. Designing scripts to guide users in modifying knowledge-based systems. In Proceedings of the Sixteenth National Conference on Artificial Intelligence. AAAI Press.
23. Gil, Y. 1994. Knowledge refinement in a reflective architecture. In Proceedings of the Twelfth National Conference on Artificial Intelligence.