An interface for targeted collection of common sense knowledge using a mixture model

Robert Speer, Jayant Krishnamurthy
MIT CSAIL
Cambridge, MA USA
{rspeer, jayant}@mit.edu

Catherine Havasi
Brandeis University
Waltham, MA USA
havasi@brandeis.edu

Dustin Smith, Henry Lieberman, Kenneth Arnold
MIT Media Lab
Cambridge, MA USA
{dustin, lieber, kcarnold}
@media.mit.edu

ABSTRACT

We present a game-based interface for acquiring common sense knowledge. In addition to being interactive and entertaining, our interface guides the knowledge acquisition process to learn about the most salient characteristics of a particular concept. We use statistical classification methods to discover the most informative characteristics in the Open Mind Common Sense knowledge base, and use these characteristics to play a game of 20 Questions with the user. Our interface also allows users to enter knowledge more quickly than a more traditional knowledge-acquisition interface. An evaluation showed that users enjoyed the game and that it increased the speed of knowledge acquisition.

Author Keywords
Common sense reasoning, human computation, hierarchical Bayes model, knowledge acquisition

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces—*Natural Language*

INTRODUCTION

Ask any typical user, and they might tell you that computers are frustrating to interact with because they just don't have any common sense. If computer interfaces were able to understand the basic facts that people already know, they would be interact with users more effectively and naturally. Applications can take advantage of these facts to make reasonable assumptions that improve the user experience. An application that interacts with our daily lives should have a representation of our goals, our problems, and the surroundings in which we live; an application that deals with natural language should be able to understand which words make sense in context. In order for these intelligent computer interfaces to see the world from the perspective of their users, they must have access to a wealth of information about the world that human users take for granted. This information, which people often take for granted and is not frequently represented in computers, is common sense knowledge.

Much of the recent progress in common sense inference has focused on filling in the gaps in large, existing knowledge bases [28, 14]. While originating in English, these large common sense knowledge-bases increasingly have been expanding to other languages [1][8][9] and using new techniques [7] to incorporate external, often domain specific, information. For these applications, it is important to be able to add a new concept to an existing knowledge base as quickly and effectively as possible. Such targeted knowledge acquisition is focused on integrating a single concept into an existing body of knowledge.

In this paper, we improve the targeted data acquisition process by using a model that represents the known concepts in a hierarchy of clusters. A beta-binomial mixture model discovers the clusters and the features that define them. We use this model to build an intelligent interface that plays a game of 20 Questions with the user. This game serves the dual purpose of motivating volunteer contributions and increasing the throughput of new knowledge when interacting with a user.

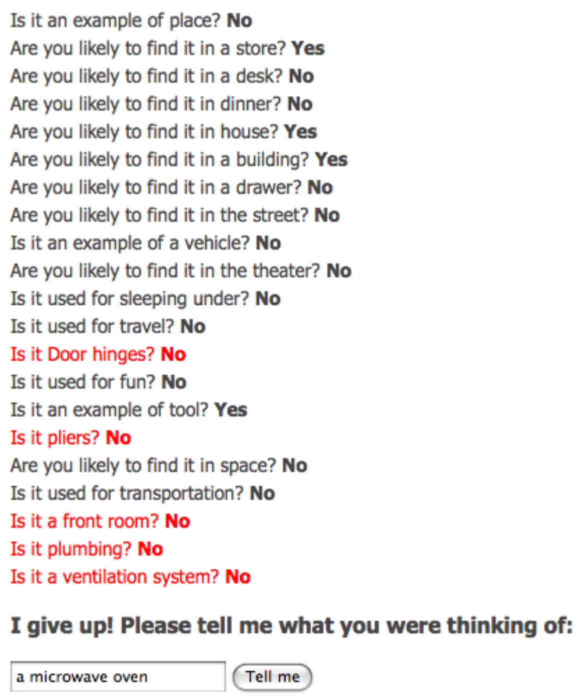
The symbiotic relationship between user interfaces and common sense

When people interact with each other, that interaction is mediated by contextual and preexisting knowledge about the situation, about how people think and react in similar situations and by general world knowledge. This complex web of knowledge, which is so second nature in human to human interactions, is missing when a human interacts with a computer interface.

Accumulating common sense knowledge has been recognized as a critical sub-goal for AI [18]; and has also proven useful for nearer-term problems such as activity recognition [23, 33], planning [5], sentiment analysis [15], speech recognition [11], and other innovative language processing applications [12].

Before humans can enjoy the benefits of intelligent user in-

Submitted for review to IUI 2009.


Is it an example of place? **No**
Are you likely to find it in a store? **Yes**
Are you likely to find it in a desk? **No**
Are you likely to find it in dinner? **No**
Are you likely to find it in house? **Yes**
Are you likely to find it in a building? **Yes**
Are you likely to find it in a drawer? **No**
Are you likely to find it in the street? **No**
Is it an example of a vehicle? **No**
Are you likely to find it in the theater? **No**
Is it used for sleeping under? **No**
Is it used for travel? **No**
Is it Door hinges? **No**
Is it used for fun? **No**
Is it an example of tool? **Yes**
Is it pliers? **No**
Are you likely to find it in space? **No**
Is it used for transportation? **No**
Is it a front room? **No**
Is it plumbing? **No**
Is it a ventilation system? **No**

I give up! Please tell me what you were thinking of:

a microwave oven

Figure 1. Open Mind learns facts about the concept "microwave oven" from a session of 20 Questions.

terfaces enriched with common sense, some must share their own with computers. Despite advances in machine learning, the most reliable way to acquire common sense knowledge is to solicit it directly from people. If we aim to collect knowledge from people, it is important to create a user interface that makes it easy and enjoyable for users to find and fill gaps in knowledge.

The common sense acquisition problem

Our typical daily experiences draw from vast stores of common sense knowledge. For example, when we enter an unfamiliar restaurant or store, we use many common sense assumptions to manage this new experience. The assumptions range from the knowledge that money can be exchanged for goods and services to the conviction that the floor we stand on will support our weight. Conversations between people similarly depend on these basic, unspoken assumptions. Grice's theory of pragmatics [4] states that when communicating, people avoid providing obvious information. Hence, common sense information is usually not written down, and we must make an effort to collect it. Two such efforts are Cyc and the volunteer-driven Open Mind Common Sense (OMCS)[26] projects. In the last five years, both of these projects have realized advanced and intelligent user interfaces are required to retain volunteer contributers.

In 2005, the Cyc project developed tools [32] to enable lightly trained volunteers to enter or confirm information. Two of the main components of knowledge in Cyc are its "ground facts" (GAFs) and rules. Because rules are hard to learn, Cyc focuses on the acquisitions of GAFs. One of the developed tools, Factivore, is for acquiring domain specific knowledge — an example provided was of restaurants in the Austin area. Factivore significantly increased the entry speed of GAFs but had the disadvantage that ontologists must hand-prepare each of Factivore's frames. The other tool, "Predicate Populator", looks at web pages and suggests words that fit in Cyc category frames which frequently co-occur with a given word. This tool also required significant ontologist preparation.

Games for human-based computation

How do we motivate users to contribute knowledge? In an early version of OMCS, our survey uncovered that prolific contributors were most motivated to contribute knowledge by a sense of "interaction" and the impression that the computer was learning or understanding their entries [6, 3]. Consequently, the new interface for OMCS[1] was designed with interactivity in mind [27].

In the past few years, Luis von Ahn and colleagues have demonstrated that a large audience of people could be enticed into contributing knowledge by playing a web-based game. This labor-for-entertainment approach [30] is sometimes called Human Computation.

Two examples of such games are Peekaboom and the ESP game. In the ESP game [29], a player is presented with an image from the Internet which needs to be labeled. Labeling images is useful, both for Internet image search and for aiding those with disabilities. Two players who are unknown to each other are shown an image. Each must "guess" what the other player is thinking by entering words which describe the object in the picture. When the two players enter the same word, they are scored by how long it took them to agree. The words are often good labels since users tend to enter the most general descriptions for images hoping to match with their fellow player. The Peekaboom [31] game is a more advanced version of the same principles — pairs of users attempting to label components of images.

Verbosity [17] is a general common sense knowledge acquisition game. Like the games above, Verbosity is a game for two players. One of these players is given a word and must get their partner to guess the word. That player can fill in Open Mind-like templates with any word that is not the provided word and the second user must guess the word from the clues. For example, if the word is "horse" one could say "it has a hoof".

One common sense knowledge acquisition game, Common Consensus [13], is similar to the television game show Family Feud, in which contestants compete to give the *most common* responses to a prompt (*e.g.,*"things you would find in a car"). Rather than polling the audience to predetermine valid answers, Common Consensus computes "commonality" on-the-fly based on all of the online contestants' responses.

The Restaurant Game [22] was designed to make computer game AIs better able to model human social interactions. In this two-player game, one user plays the diner and an-

[1]`http://openmind.media.mit.edu`

other plays the waitress. Their objective is to complete a normal dining experience by interacting with each other, using the objects in the environment. The data collected is used to train an intelligent Plan Network which will become the game AI for a single-player restaurant game.

Robin Burgener's commercial 20 Questions game, "20Q", [2] is a web-based game (also packaged as a retail toy) that tries to guess what object you're thinking of. This game is powered by a neural network which was trained over a period of years. The biggest differences between this game and ours are all derived from the difference in purpose between our game and Burgener's: our game is intended to improve knowledge acquisition and inference, and his is focused on entertainment alone. Burgener's game works with a fixed set of questions (the Web site version can expand its set of concepts), while we are interested in a learning method that grows with our data set in both its concepts and the features it can apply to them. Also, Burgener's game is optimized for the task of playing the game well and narrowing down the choices to a single concept in as few questions as possible, while our objective is to determine the object's similarity to other known concepts. For us, playing the game optimally is a secondary concern. Regardless of these differences, we have drawn inspiration from the web-based game in parts of our design, such as the constantly-updated list of potential concepts.

Von Ahn stressed [30] that games are a useful means of knowledge acquisition when the problem in question is one that is hard for computers and easy for humans. A user who chooses to enter a novel concept into the Open Mind system is probably knowledgeable about the concept. In most cases, this basic common sense about the object cannot be found by automatically mining corpora.

Our 20 Questions game produces natural language questions based on the knowledge that is already present in Open Mind. The questions are intended to identify a sufficiently small cluster of objects that the object in question can be compared to, and in some questions the game can even guess what the object is. Interestingly, though, the interactions where the game guesses correctly are the least useful for acquiring new knowledge. If the game can guess a concept based on the user's answers to the questions, then Open Mind already has enough information to determine what cluster that object belongs in. Learning and winning the game are goals that are generally not accomplished at the same time.

A session with the game, along with the user's answers, appears in Figure 1.

REPRESENTATIONS FOR COMMON SENSE

There is no consensus on a universal way to represent semantic content, although diverse proposals exist [24, 21, 25, 10]. Instead of committing to a particular representation, the creators of OMCS separated the source data from the machine-interpretable resources that can be derived from it. The source data is stored in natural language, and is often marked up to identify key concepts. An example of an assertion in OMCS is "You are likely to find *milk* in *a refrigerator*". These sentences must be reformulated to be used in machine learning. Computer-friendly reformulations of the OMCS data include the semantic network ConceptNet [16, 8], and a factorized matrix of the data's principal components, AnalogySpace [28].

Our model builds on the concept/feature representation that is derived from ConceptNet's data model. Assertions in ConceptNet are characterized by three components: two *concepts* and a *relation* between them. Concepts represent sets of closely related phrases of natural language, so that "a dog", "dogs", and "many dogs" are the same concept. *Relations*, such as *UsedFor* or *PartOf*, summarize the connection between concepts.

The concept/feature representation, used by AnalogySpace and the clustering model, expresses all the assertions using a single relation between concepts and *features*; this allows the knowledge base to be represented as a sparse matrix, and simplifies the algorithms required to learn from it. A feature is a pairing of a concept with a relation which forms a complete assertion when combined with a concept. For example, "______ is part of a house" is a feature of the concept "door". Specifically, it is a *right feature*, because the feature appears to the right of the free concept; there are also *left features* such as "A door is part of ______".

Features are a particularly useful representation in the game of 20 questions, because a feature contains exactly the information required to ask a question. For example, the feature "______ is part of a house" can be reformulated as the question "Is it a part of a house?"

AnalogySpace

OMCS has recently used dimensionality reduction through singular value decomposition (SVD) as a method of reasoning, which has the benefits of smoothing the existing knowledge and proposing new assertions that fill in the gaps [27]. When a truncated SVD is run on the concept/feature representation of ConceptNet, the result is a vector space representation called AnalogySpace. The effect of this dimensionality reduction is to transfer sparse information between concepts using induction over their features.

The input to the SVD assigns a "truth score" to each entry in the concept/feature matrix. This score does double duty as a truth value and a confidence score. A predicate's score is zero if no information is known about that predicate. If the predicate is considered true in ConceptNet, the score is a positive number, indicating the number of people who have verified that assertion. If it is considered false, the score is the *negative* of the number of people who have verified the assertion. The scores form something similar to a Gaussian distribution, except that it has considerably more weight on the positive side (only 3.5% of ConceptNet predicates are negative).

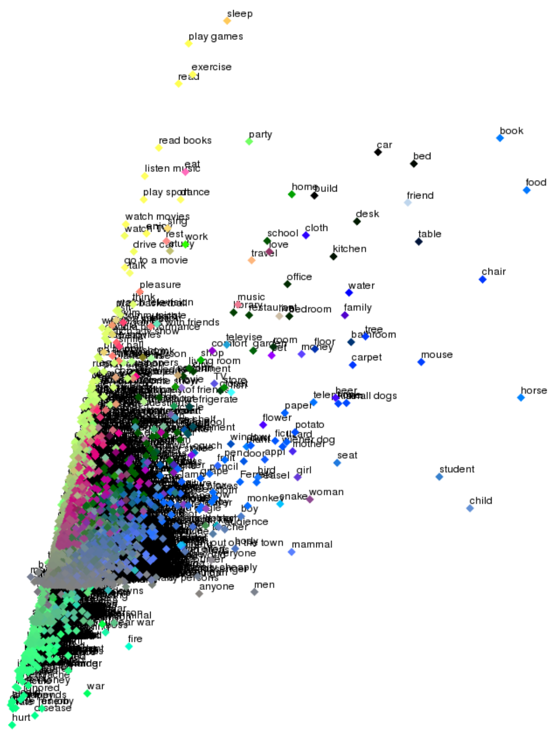


Figure 2. A 2-D projection of a part of AnalogySpace. Note the distinguishable clusters of things that are enjoyable (upward), things that are disliked (downward), and things that are animate (rightward).

This matrix of scores is normalized, so that each concept is now described by a Euclidean vector of length 1 over the space of features, and then this sparse data is run through a truncated singular value decomposition that retains only the strongest k principal components. In this task, we chose $k = 50$.[2] This gives a 50-dimensional space in which both concepts and features can be represented and compared to each other. In this space, similarities between concepts or features can be revealed by the cosine similarity of their vectors.

In particular, a high cosine similarity between a concept and a feature indicates a *prediction* that the feature is likely to apply to the concept, according to the inductive "smoothing" that results from reducing the dimensionality of the data. Discovering high cosine similarities between concepts and features that were previously unconnected is a way to propose new assertions that may be true. This technique is used on the main OMCS web site to generate relevant yes/no questions to ask the users.

Similar concepts and properties form clusters that point in the same direction away from the origin, possibly with different magnitudes, leading to a "spiky" overall shape of ConceptNet. However, no previous inference process over ConceptNet takes this structure into account. Dimensionality reduction using SVD assumes that the data forms a smooth Gaussian in its lower-dimensional space. One result of this inaccurate assumption is that AnalogySpace gives poor predictions when not many things are known about a concept.

When teaching Open Mind about a new concept, we need to take into account the fact that some areas of this vector space are more densely populated than others. This leads us to use a clustering model to classify new concepts. Instead of AnalogySpace's goal of filling in the gaps in what the system already knows, the goal of this model is to ask questions that sort out what kind of thing is being discussed, in order to find out what other questions would be reasonable to ask. This is much like the goal of the game "Twenty Questions", so this is how we have presented it in our user interface.

CLUSTERING AND LEARNING A CLASSIFICATION TREE

The beta-binomial mixture model

The goal of this clustering model is to place the new concept near a cluster of existing concepts, at which point AnalogySpace is able to take over and make good predictions. In order to place the concept, we need to identify clusters of concepts with an unsupervised method. We cluster the concepts based on their feature vectors, derived from the concept/feature matrix representation discussed above.

Our clustering model assumes that concepts are generated from a number of independent clusters. Clusters are defined by vectors of probabilities over features, expressing the probability that each feature is true for concepts in that cluster.

We consider the existing data, then, to represent observations about the truth of that assertion. When multiple users have entered an assertion, or approved it, it gives us positive observations that should increase the likelihood of models in which that assertion is true. When users assert that a statement is false, it gives us a negative observation, increasing the likelihood of models in which that assertion is false.

In our generative model, we suppose that these observations come from a binomial distribution whose mean is specified by the feature probabilities in the cluster. Our prior on the probability of an assertion is a beta distribution with an expected value of 0.04, representing the approximately 4% likelihood that a randomly selected feature, applied to a randomly selected concept, would be labeled as "generally true" by a user, as determined by a user study in [27].

We chose the total weight on the beta distribution to represent 0.1 of an assertion, so that real information would remain more prominent than the prior. This prior allows negative information to have an effect, by causing the probability θ of a feature to decrease from its starting value, and allows us to distinguish the truth values of "false" and "unknown".

The generative model, then, can be expressed in terms of the positive observations $y_i^{(k)}$, the total numbers of obser-

[2]The choice of k is arbitrary and represents a tradeoff between the ability to make fine distinctions and computation time. If we followed the rule of thumb that a truncated SVD should represent 95% of the variance, we could use over 400 dimensions, but this would overburden our Web application by greatly increasing both the computation time and storage space required to serve a page.

vations $n_i^{(k)}$, the cluster means $\mu i, j$, the class assignments $z^{(k)}$, the probability weights of each class π, the hyperparameters $\alpha = 0.004$ and $\beta = 0.096$ that establish the beta prior, and some theoretical parameters that are ultimately disregarded by our implementation. In statistical notation, our model is:

$$\begin{aligned}
\mu_{i,j} &\sim \text{Beta}(\alpha, \beta) \\
\pi &\sim \text{Dirichlet}(\alpha_\pi) \\
z^{(k)} &\sim \text{Multinomial}(\pi) \\
n_i^{(k)} &\sim \text{Poisson}(\gamma) \\
y_i^{(k)} &\sim \text{Binomial}(\mu_{i,z^{(k)}}, n_i^{(k)})
\end{aligned}$$

We implement this model with an expectation maximization process, incrementally determining the likelihood $Z_{i,c}$ that each concept i is in each cluster c according to the binomial distribution, and then updating each cluster's feature probabilities (denoted $\theta_{c,j}$, where c specifies the cluster and j specifies the feature) to match the concepts it is likely to contain. The process in detail is:

```
Repeat until convergence:
    For each concept $i$ and class $c$:
        $Z_{i,c} \leftarrow \prod_j \theta_{c,j}^{y_{i,j}} (1 - \theta_{c,j})^{n_{i,j} - y_{i,j}}$
    For each class $c$ and feature $j$:
        $\theta_{c,j} \leftarrow \sum_i Z_{i,c} \cdot \frac{y_{i,j}}{n_{i,j}}$
```

This algorithm requires a fixed number of clusters. We do not claim to know the "correct" number of clusters in our data, or to expect EM to correctly discover all of the clusters we want simultaneously. Instead, we use a hierarchical clustering process in which clusters can be divided into subclusters using the same generative model.

We choose to divide the data into 5 clusters at each step in this process; we arrived at the branching factor 5 by experimentation. We previously tried the minimum branching factor of 2, but found that the resulting model was ineffective at distinguishing clusters. Branching into more clusters at each step increases the time and space requirements of the algorithm for what seem to be diminishing returns.

Once EM has converged, we find the maximum-likelihood assignment of concepts to clusters, placing each concept solely in the cluster that is most likely to contain it. We then run EM again on each cluster separately, to find subclusters within that cluster. This gives us a tree of clusters which we can expand in priority order, always choosing to expand the leaf with the highest probability until the hierarchy grows sufficiently large.

Distinguishing clusters with questions

What we want to find are the questions that most effectively steer a new concept toward the populated areas of AnalogySpace, which we accomplish by putting the concept in a cluster. The questions we want to ask, then, are the questions that are best at distinguishing different clusters at various levels of the hierarchy.

First, we need to reverse the probabilities in the θ matrix using Bayes' rule. We know the probability of each feature f given each cluster c, but determining which feature to ask about requires finding the probability of each cluster given each feature.

$$p(c|f) = \frac{p(f|c)p(c)}{p(f)}$$

$$p(c|\neg f) = \frac{p(\neg f|c)p(c)}{p(\neg f)}$$

We also need to do this in the presence of already known information. If we have some set of features or their negations $\mathbf{f}_k$ that we already know, we can include them in this conditional probability. (So far, we make the simplifying assumption that features are conditionally independent of each other given a class; we are considering using a model that relaxes this assumption.)

$$\begin{aligned}
p(c|f, f_k) &= \frac{p(f, f_k, c)}{p(f, f_k)} \\
&= \frac{p(f|c)p(f_k|c)p(c)}{p(f, f_k)}
\end{aligned}$$

In practice, we leave out the denominators until we reintroduce them as normalizing factors at the end. Thus, taking known features into account requires only multiplying by their likelihood for each class, $p(f_k|c)$.

Finally, we calculate the disorder that results from asking about each feature. Omitting the f_k terms, this is:

$$-\sum_c \left(p(c) \sum_f p(f)p(c|f) \lg p(c|f) + p(\neg f)p(c|\neg f) \lg p(c|\neg f) \right)$$

The feature with the least disorder is the one that best distinguishes the classes from each other, so that is the feature that we turn into a question to ask the user. As an example, at the top level of the hierarchy, the most informative question in the absence of any other information is "Is it a kind of place?".

Drilling down through the hierarchy

In order to take advantage of hierarchy of clusters we have created, the process that generates questions must be able to choose paths that drill down through this hierarchy, asking increasingly specific questions until it identifies a small cluster of possible concepts.

When the probability $p(c|f_k)$ that the concept is in a particular cluster based on the known information increases above 0.5, the questioner tentatively selects that cluster, and begins asking questions that will distinguish its subclusters. This drilling-down process is re-calculated from the top of the hierarchy after every question, and it frequently happens that the path changes based on new information. If the likelihood of a previously chosen cluster decreases below 0.5, for example, the questioner will back up and ask more questions at the level above.

Avoiding sparsity with dimensionality reduction

The model described in the previous section performs poorly when confronted with the actual data in OMCS, because that data is too sparse. Every question it chooses to ask tells it nothing about the vast majority of concepts, because the majority of concept/feature pairs are unknown.

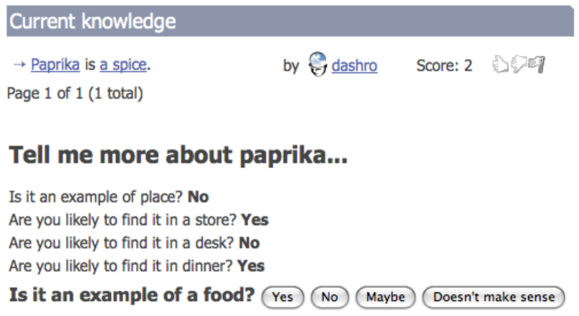
There already exists a technique for filling in the gaps, and that is to smooth the data using dimensionality reduction. Instead of training the model on just the positive and negative observations that come from users, we use virtual positive or negative observations that we generate from the AnalogySpace representation. If AnalogySpace predicts the value 0.2 for a particular combination of a concept and a feature, for example, then we consider this as 0.2 of a positive observation for that assertion. As a result, every combination of a concept and a feature has *some* data that can be used when asking questions, even if it only counts for a small fraction of an observation.

A side effect of this is that we can no longer represent assertions with both positive and negative observations, because the smoothing produces a single value for every assertion that is either positive or negative. Each $y_{i,j}$ is either 0 or equal to $n_{i,j}$. This slightly misrepresents the probability of "controversial" assertions, treating them as if they are observed fewer times instead of being observed with opposite values, but it is unclear whether this has a meaningful effect on the results.

When we smooth the input data in this way, the probability of every concept can be affected by every question, and this greatly decreases the number of questions required to identify a known concept. Even though the majority of concepts are not asserted to be either "places" or "not places", the model can adjust the probability of those concepts after seeing the answer to "Is it a kind of place?" by determining whether they are sufficiently similar (or dissimilar) to concepts that are places.

INTERFACE DESIGN OBJECTIVES

The "20 Questions" interface is quite simple, so in addition to being a game that users can use to teach Open Mind, it can also be incorporated into the main Open Mind interface unobtrusively. When a user visits the page for a concept that contains few or no assertions, the site will present that user with the broad questions generated from the mixture model instead of the inferences from AnalogySpace. In this situation, Open Mind is not trying to guess what the concept is, so this mode of interaction does not get the benefit of being a game, but it remains an efficient way of collecting the basic knowledge that will allow Open Mind to go on to use other forms of inference. An example of this interface appears in Figure 3.


Current knowledge

→ Paprika is a spice. by dashro Score: 2

Page 1 of 1 (1 total)

Tell me more about paprika...

Is it an example of place? **No**
Are you likely to find it in a store? **Yes**
Are you likely to find it in a desk? **No**
Are you likely to find it in dinner? **Yes**

Is it an example of a food?

Figure 3. Using the 20 Questions interface to develop a concept.

In constructing the interface we followed four main design principles.

Feedback

As mentioned above, we've found that interactivity is a critical component in reducing user attrition. If users feel they are teaching the system and interacting with it, they will enter more data and be more satisfied with their experience. This is the motivation behind a feature of the interface that shows the user which concepts are the computer's current best guesses: the user can directly observe how their answers are affecting the computer's decisions.

User enjoyment

The game should be more fun for the users, and better at retaining users, than a static data entry task.

Minimalism

The interface should be usable not just as a stand-alone game, but as a tool for learning that can be used elsewhere in Open Mind. It is important that the game does not further clutter the OMCS web site, and that the user sees it as a natural way to interact with the page in which it is embedded, which could be a summary page that describes a particular concept, or a page for translating a concept into another language. This motivated us to design it with a minimal interface. The only things that appear besides the text and buttons that are necessary to play the game are a list of previous statements (giving continuity to the user experience) and the small list of predicted concepts (providing feedback).

Effortless Acquisition

In the standard Open Mind interface, few users would sit down and enter 20 statements in a row, regardless of whether they are positive or negative. It is an important aspect of the 20 Questions interface that each incremental contribution of knowledge requires very little effort. Users can teach a significant amount of useful knowledge to Open Mind without focusing on the fact that they are doing so, because they are focused instead on the game.

Sample Interaction

The user begins an interaction with our 20 questions game by mentally selecting a concept to enter information about. This concept can range from a concrete object, such as “pen”, to an activity like “playing hockey”, to an abstract idea such as “liberty”. The system then presents the user with a question about his or her mentally-selected concept, expressed in natural language. Each question has four possible responses: “Yes”, “No”, “Maybe”, and “Doesn’t Make Sense”. Once the user answers the question, the system selects another question to ask the user. (The system generates each question by choosing the lowest-disorder feature from the hierarchical classifier, given the responses to previous questions. These questions tend to narrow down the likely choices for the user’s concept.) Some frequently asked questions include “Is it a place?” and “Is it used for fun?”.

Throughout the interaction, the system presents the user with a list of “possible concepts”. These concepts are the system’s current best guesses for the user’s concept. As the system collects more responses to its questions, these guesses are refined. The possible concepts are computed using the product of the result in the hierarchical classifier and the score assigned by AnalogySpace. Both AnalogySpace and the hierarchical classifier produce likelihood estimates for every concept, and these likelihood estimates are multiplied by each other to produce the list of likely concepts. We combine the estimates because each system excels at predicting concepts at different levels of granularity; the classifier places us in the right cluster, and AnalogySpace finds the concept within the cluster.

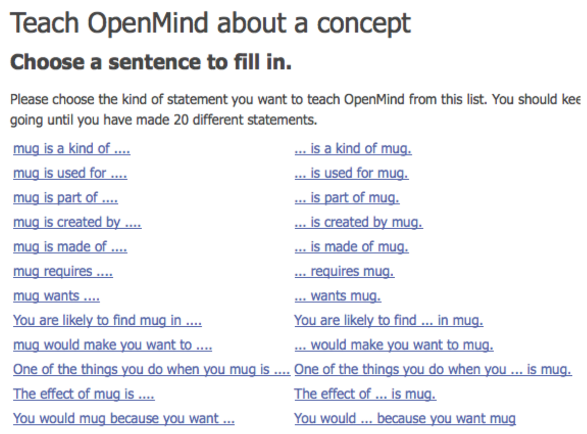
The system attempts to guess the user’s concept after collecting 12, 15, and 18 pieces of information. It guesses once after the 12th and 15th pieces of information, and 3 times after the 18th. The system always guesses the current most likely concept (derived in the same manner as the possible concepts). If the system guesses correctly, the game ends. Otherwise, the game continues until the system asks 20 questions. At this point, the user is prompted to enter the concept he or she was thinking of, and the game ends.

EVALUATION

User Test Design

We tested our interface with users on the Internet to determine if it gathered knowledge more effectively than the current OpenMind interface. Upon starting the user test, users were assigned to evaluate one of two interfaces at random.

The first interface, designed to mimic the previously existing method for teaching OpenMind about a new concept, had the user select a concept and manually input 20 facts about it. First the user was asked to enter a concept of their choice. Then, the interface presented the user with a choice of sentence frames involving their concept to fill in. Figure 4 shows the list of sentence frames that appears when the selected concept is “mug”. After choosing a frame, the user is asked to complete it using the standard OMCS interface, as seen in Figure 5.



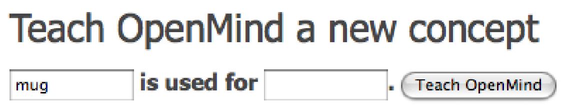
Teach OpenMind about a concept

Choose a sentence to fill in.

Please choose the kind of statement you want to teach OpenMind from this list. You should kee
going until you have made 20 different statements.

mug is a kind of	... is a kind of mug.
mug is used for	... is used for mug.
mug is part of	... is part of mug.
mug is created by	... is created by mug.
mug is made of	... is made of mug.
mug requires	... requires mug.
mug wants	... wants mug.
You are likely to find mug in	You are likely to find ... in mug.
mug would make you want to	... would make you want to mug.
One of the things you do when you mug is	One of the things you do when you ... is mug.
The effect of mug is	The effect of ... is mug.
You would mug because you want ...	You would ... because you want mug

Figure 4. The first user test interface, designed to mimic the Open Mind interface. After entering a concept, the user is presented with a list of possible assertions involving the concept.

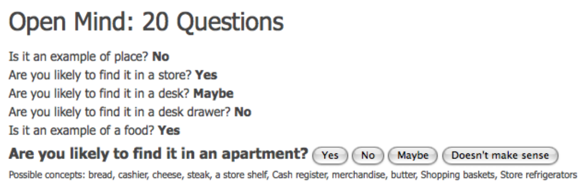


Teach OpenMind a new concept

mug **is used for** ______ . Teach OpenMind

Figure 5. The traditional, frame-based interface for entering an assertion.

The second interface played a game of 20 questions with the user, interactively choosing questions to ask based on the user’s previous responses. An example of this for the concept “apple” can be seen in figure 6. During this process, the user is shown a list of the system’s current guesses for the user’s concept.



Open Mind: 20 Questions

Is it an example of place? **No**
Are you likely to find it in a store? **Yes**
Are you likely to find it in a desk? **Maybe**
Are you likely to find it in a desk drawer? **No**
Is it an example of a food? **Yes**
Are you likely to find it in an apartment? Yes No Maybe Doesn't make sense
Possible concepts: bread, cashier, cheese, steak, a store shelf, Cash register, merchandise, butter, Shopping baskets, Store refrigerators

Figure 6. The twenty questions user test interface. The concept for the game currently in progress is “apple”.

Survey

After using either interface, users were shown the list of positive statements that they had contributed to Open Mind, and then filled out a survey about their experience. The survey asked each user whether they agreed with the following statements (phrased as a mixture of positive and negative statements to encourage respondents to think about them before clicking):

1. I enjoyed this activity.
2. This was not an efficient way to enter information.
3. I felt like I was teaching the system.
4. The system was adapting as I entered new information.

5. I found this interface unintuitive.
6. I would use this system again.

All of these questions were scored on a 5-point Likert scale. We also asked users if they had used OpenMind before and recorded how long it took users to complete the task. We used anonymous cookies to avoid giving the survey to users who went back and took the user test again.

Results

106 people participated in our user test and responded to the survey. We present their responses in graphical form in Figure 7 and Figure 8.

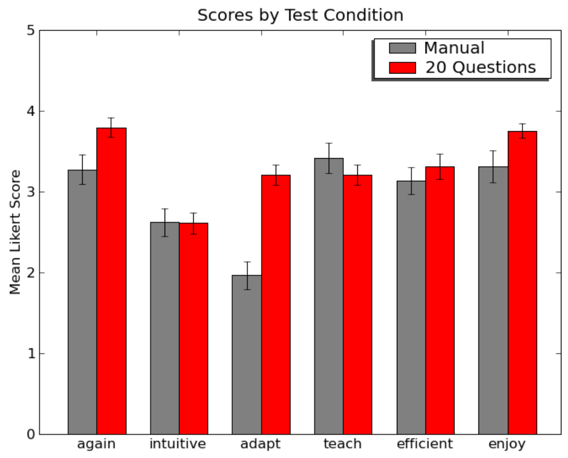


Figure 7. The mean and SEM of the survey responses, grouped by test condition.

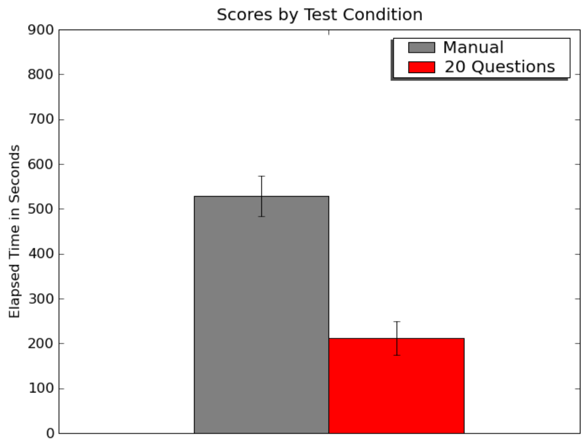


Figure 8. The mean and SEM of the elapsed time to complete each task.

We used a t-test to measure the statistical significance of our survey results. We found three statements on which 20 Questions outperformed the other condition: the differences between the means of the responses to “I would use this system again”, “I enjoyed this activity”, and “The system was adapting as I entered new information” were statistically significant at the $p = 0.05$ level. We also found that the difference between the completion times to be significant at the $p = 0.05$ level. In our calculations of completion times, we eliminated any users who took over 1 hour to complete the task, based on the assumption that the user simply left his or her browser window open.

The results of our user test show that users are capable of entering knowledge more quickly with our system. In addition, users claimed they enjoyed interacting with the 20 questions system and were more likely to use the 20 questions system again. These results indicate that the 20 questions interface will help us engage users and acquire more common sense knowledge.

DISCUSSION

In his books *Society of Mind* [19] and *The Emotion Machine* [20], Marvin Minsky discusses the common sense reasoning problem and, indeed, the entire problem of knowledge representation. Minsky coins the word “panalogy” to refer to switching representations in order to better reason about a problem, saying that being able to switch rapidly between representations is an important part of reasoning. Minsky cites this as a way to enable our reasoning systems to become “unstuck” when they encounter a difficult problem. It is important, then, that our reasoning not become fixed within a method or domain; we should be able to fluidly transfer between them.

Our interface is an example of using multiple representations to solve a single problem. When teaching Open Mind about a new concept, the hierarchical clustering model allows us to efficiently learn general information about the concept. These facts let us place the new concept within a cluster of closely-related concepts. Once within that cluster, we can switch representations and use AnalogySpace to learn details about the concept. These two knowledge representations complement each other and help our system to learn more efficiently.

Another advantage of the interface is that it collects both positive and negative assertions from users. Negative assertions seem to be under-represented in the Open Mind Common Sense, appearing much less frequently than positive assertions. Negative assertions are important because they increase our ability to distinguish what is false from what we simply do not know. Currently, data sparsity prevents us from distinguishing untrue assertions from unknown assertions. The negative assertions contributed by 20 questions will increase our ability to make these distinctions in the future.

We believe that this interface will encourage more people to contribute common sense knowledge, while making it easy and straightforward for them to contribute knowledge that is generally useful. By framing common sense acquisition as an entertaining game, we encourage user participation and keep the interest of our existing users. We believe this game will increase both the appeal and the effectiveness of Open

Mind Common Sense.

ADDITIONAL AUTHORS

REFERENCES

1. J. Anacleto, H. Lieberman, M. Tsutsumi, V. Neris, A. Carvalho, J. Espinosa, and S. Zem-Mascarenhas. Can common sense uncover cultural differences in computer applications? In *Proceedings of IFIP World Computer Conference*, Santiago, Chile, 2006.

2. S. Cass. Holiday gifts. *IEEE Spectrum*, 42(11), 2005.

3. D. Garcia, C. Havasi, and K. Todd. Interactively improving the OpenMind database: A game for interactive OpenMind inprovement (GIOMI). Media lab working paper, Massachusetts Institute of Technology, 2002.

4. P. Grice. Logic and conversation. In *Speech Acts*. Academic Press, 1975.

5. R. Gupta and K. Hennacy. Commonsense reasoning about task instructions. *AAAI-05 Workshop on modular construction of human-like intelligence. Pittsburgh, PA, July*, 10:05–08, 2005.

6. C. Havasi. Open Mind user test survey results. Personal communication, 2002.

7. C. Havasi. Discovering semantic relations using singular value decomposition based techniques. *To appear at the NSF Symposium on Semantic Knowledge Discovery, Organization and Use*, 2008.

8. C. Havasi, R. Speer, and J. Alonso. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, Borovets, Bulgaria, September 2007.

9. Y. Jung, J.-Y. Lee, Y. Kim, J. Park, S.-H. Myaeng, and H.-C. Rim. *Computational Linguistics and Intelligent Text Processing*, pages 23–34. Springer, Heidelberg, 2007.

10. D. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 11:33–38, 1995.

11. H. Lieberman, A. Faaborg, W. Daher, and J. Espinosa. How to wreck a nice beach you sing calm incense. *Proceedings of the 10th international conference on intelligent user interfaces*, Jan 2005.

12. H. Lieberman, H. Liu, P. Singh, and B. Barry. Beating common sense into interactive applications. *AI Magazine*, 25(4):63–76, 2004.

13. H. Lieberman, D. Smith, and A. Teeters. Common consensus: A web-based game for collecting commonsense goals. *Workshop on Common Sense for Intelligent Interfaces ACM . . .* , Jan 2007.

14. T. Lin. Analogical inference over a common sense database. In *Eighteenth national conference on Artificial intelligence*, pages 955–956, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

15. H. Liu, H. Lieberman, and T. Selker. A model of textual affect sensing using real-world knowledge. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 125–132, New York, NY, USA, 2003. ACM.

16. H. Liu and P. Singh. ConceptNet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, Sep 2004.

17. M. K. Luis von Ahn and M. Blum. Verbosity: A game for collecting common-sense knowledge. In *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI*, 2004.

18. J. McCarthy. Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, London, 1959. Her Majesty's Stationary Office.

19. M. Minsky. *Society of Mind*. Simon & Schuster, March 1988.

20. M. Minsky. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, November 2006.

21. E. T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann, 2006.

22. J. Orkin and D. Roy. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development*, 3(1), 2007.

23. W. Pentney, A. Popescu, S. Wang, and H. KAUTZ. Sensor-based understanding of daily life via large-scale use of common sense. *Proceedings of AAAI*, Jan 2006.

24. J. Pustejovsky. *The Generative Lexicon*. MIT Press, Cambridge, MA, 1998.

25. L. K. Schubert. Can we derive general world knowledge from texts? In *Proceedings of the Human Language Technology Conference*, pages 24–27, San Diego, CA, 2002.

26. P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu. Open Mind Common Sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1223–1237, London, UK, 2002. Springer-Verlag.

27. R. Speer. Learning common sense knowledge from user interaction and principal component analysis. *MS Thesis. Massachusetts Institute of Technology*, page 110, Aug 2007.

28. R. Speer, C. Havasi, and H. Lieberman. AnalogySpace: Reducing the dimensionality of common sense knowledge. *Proceedings of AAAI 2008*, July 2008.

29. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI*, 2004.

30. L. von Ahn and L. Dabbish. General techniques for designing games with a purpose. *Communications of the ACM*, August, 2008.

31. L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64, New York, NY, USA, 2006. ACM.

32. M. Witbrock, C. Matuszek, A. Brusseau, R. Kahlert, C. B. Fraser, and D. Lenat. Knowledge begets knowledge: Steps towards assisted knowledge acquisition. In *AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors*, 2005.

33. D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. *Proc. of AAAI*, Jan 2005.