Digital Intuition: Applying Common Sense Using Dimensionality Reduction

Catherine Havasi and James Pustejovsky, *Brandeis University*

Robert Speer and Henry Lieberman, *Massachusetts Institute of Technology*

To help AI systems acquire commonsense reasoning, OMCS provides a broad collection of basic knowledge and the computational tools to work with it.

When we encounter new situations, such as entering an unfamiliar restaurant or store, we rely on our background knowledge to act and communicate appropriately. This background knowledge includes, for example, the knowledge that you can pay money for goods and services and the conviction that the floor will hold your weight. When people communicate with each other, they rely on similar background knowledge, which they almost never state explicitly. This follows from the maxim of pragmatics that people avoid stating information that is obvious to the listener.[1]

In the absence of a learning system as complete as the human brain, automatically acquiring all this frequently unstated knowledge would be difficult. But for an AI system to understand the world that humans live in and talk about, it needs to have this unspoken background knowledge. It needs a source of information about the basic relationships between things that nearly every person knows. In one way or another, this implicit knowledge must be made explicit so that a system can use it computationally.

The goal of the Open Mind Common Sense (OMCS) project is to provide intuition to AI systems and applications by giving them access to a broad collection of basic knowledge, along with the computational tools to work with it. This knowledge helps applications understand the way objects relate to each other in the world, people's goals in their daily lives, and the emotional content of events or situations.

Reflecting the way people change thought processes and representations to attack different problems, we designed the OMCS system to easily transition between several data formats, using the best representation for an application or problem. Our semantic network, ConceptNet, is built from a corpus of commonsense knowledge collected and rated by volunteers on the Internet. ConceptNet powers our rea-

1541-1672/09/$25.00 © 2009 IEEE
Published by the IEEE Computer Society

soning engine, AnalogySpace, which uses factor analysis to build a space representing the large-scale patterns in commonsense knowledge. It uses these patterns to reason over the collected information and infer additional common sense.

Blending is a technique that performs inference over multiple sources of data simultaneously, taking advantage of the overlap between them. This enables common sense to serve as a basis for inference in a wide variety of systems and applications. Using this natural extension of the AnalogySpace process, applications can achieve "digital intuition" about their own data, making assumptions and conclusions based on the connections between that specific data and the general common sense that people have.

Acquiring and Representing Common Sense

Open Mind Common Sense takes a distributed approach to the problem of commonsense knowledge acquisition. The project allows the general public to enter commonsense knowledge into it, without requiring any knowledge of linguistics, artificial intelligence, or computer science. (See the sidebar, "Using Open Mind Common Sense," to find out how to contribute.) The OMCS has been collecting commonsense statements from volunteers on the Internet since 2000. In that time, we've collected more than 700,000 pieces of English-language commonsense data from more than 16,000 contributors. The OMCS project has expanded to other languages, with sites collecting knowledge in Portuguese, Korean, and Japanese. (See "Other Collections of Common Sense" for related work.)

Each interface to OMCS presents knowledge to its users in natural language, and collects new knowledge in natural language as well. OMCS uses different activities to elicit many types of commonsense knowledge from contributors. Some of this knowledge is collected in free text, and some of it is collected from templates that users fill in, such as "_______ can be used to _______." Knowledge collected from these templates can be processed more easily and reliably than free text, but free text can be a useful guide for creating new templates that represent sentence patterns that our contributors have used frequently. There are 90 sentence patterns used in the creation of the ConceptNet 3.5 semantic network.

For the knowledge we collect to become computationally useful, it must be transformed from natural language into more structured forms. Much of OMCS's software is built on three interconnected representations: the natural language corpus that people interact with directly; ConceptNet, a semantic network built from this corpus; and AnalogySpace, a matrix-based representation that uses dimensionality reduction to infer new knowledge (which can then be added to ConceptNet).

We take steps to make sure that all of OMCS's knowledge, no matter what representation it's currently stored in, can be expressed in natural language. Natural language isn't something to be abstracted over and avoided; it's our system's anchor to the real world that people talk about.

ConceptNet

ConceptNet represents the information in the OMCS corpus as a directed graph.[2,3] The nodes of this graph are concepts, and its labeled edges are assertions of common sense that connect two concepts. Figure 1 shows a slice of ConceptNet surrounding the word "cake."

Concepts represent sets of closely related natural-language phrases, which could be noun phrases, verb phrases, adjective phrases, or clauses. In particular, ConceptNet defines concepts as the equivalence classes of phrases after a normalization process that removes function words, pronouns, and inflections. (As of ConceptNet 3.5, we remove inflections using a tool based on the multilingual lemmatizer MBLEM.[4]) The normalization process avoids unnecessarily sparse data and prevents some duplication of information by making phrases equivalent when they seem to have approximately the same semantics but are expressed in different ways. Using this process, for example, the phrases "drive a car," "you drive your car," "driving cars," and "drive there in a car" all become the same concept, represented with the normalized form *drive car.*

ConceptNet expresses assertions as relations between two concepts, selected from a limited set of possible relations. The various relations represent common sentence patterns found in the OMCS corpus; in particular, every fill-in-the-blanks template used on the

Using Open Mind Common Sense

To interact with Open Mind Common Sense, please visit: http://openmind.media.mit.edu.

The OMCS corpus, ConceptNet, and Divisi are all available and free to download. Divisi is a software package for reasoning by analogy and association over semantic networks, including commonsense knowledge. It contains methods for AnalogySpace-style reasoning as well as blending. These can be found at the following: http://conceptnet.media.mit.edu and http://analogyspace.media.mit.edu.

If you are interested in starting a commonsense knowledge acquisition project in your language, please contact us at conceptnet@media.mit.edu.

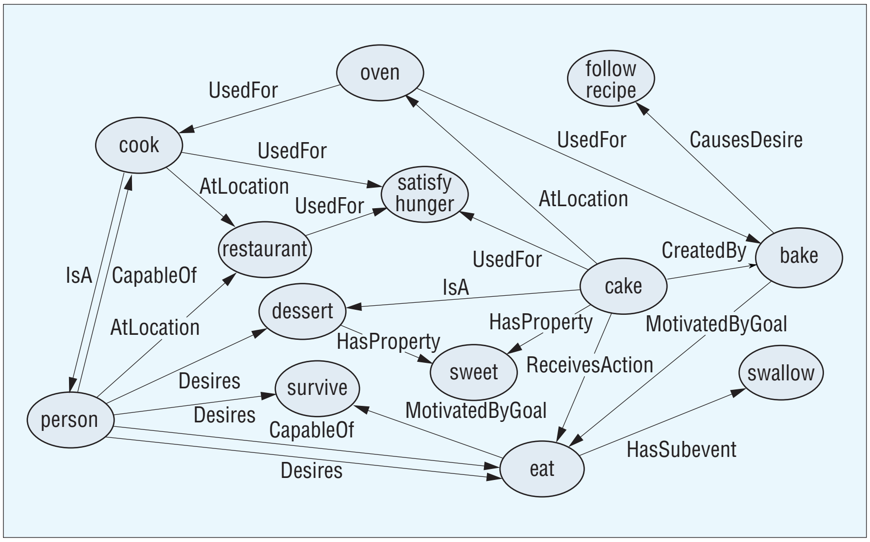

Figure 1. ConceptNet describes the common-sense relationships between natural language words and phrases. This diagram shows some of the nodes and links surrounding the concept "cake."

Table 1. The 20 relations in ConceptNet 3.5.

Relation	Example sentence pattern
IsA	NP is a kind of NP.
UsedFor	NP is used for VP.
HasA	NP has NP.
CapableOf	NP can VP.
Desires	NP wants to VP.
CreatedBy	You make NP by VP.
PartOf	NP is part of NP.
HasProperty	NP is AP.
Causes	The effect of NP|VP is NP|VP.
MadeOf	NP is made of NP.
AtLocation	Somewhere NP can be is NP.
DefinedAs	NP is defined as NP.
SymbolOf	NP represents NP.
ReceivesAction	NP can be VP (*passive*).
HasPrerequisite	Before you VP, you must VP.
MotivatedByGoal	You would VP because you want to VP.
CausesDesire	NP would make you want to VP.
HasSubevent	One of the things you do when you VP is NP|VP.
HasFirstSubevent	The first thing you do when you VP is NP|VP.
HasLastSubevent	The last thing you do when you VP is NP|VP.

AP: adjectival phrase; NP: noun phrase; VP: verb phrase. | indicates a choice between phrase types.

knowledge collection Web site is associated with a particular relation. Table 1 shows the current set of 20 relations, along with an example of a sentence pattern associated with each relation.

Each assertion is associated with a frequency value that can express whether people say the relationship sometimes, generally, or always holds; there are also frequency values that introduce negative contexts, to assert that a relationship rarely or never holds. Independently of the frequency, assertions also have a score representing the system's confidence in that assertion. When multiple users make the same assertion independently, that increases the assertion's score; users can also choose to increase or decrease an assertion's score by rating it on the OMCS Web site. This allows collaborative filtering of deliberate or inadvertent errors.

To create ConceptNet from the natural-language assertions in OMCS, we use a shallow parser to match the assertions against patterns such as the ones shown in Table 1. For example, the sentence, "You make an apple pie by baking it," becomes the assertion represented internally as (*apple pie*, CreatedBy, *bake*); Figure 2 provides an illustration.

The representation of ConceptNet described so far simplifies the contributed natural-language text into a more computationally useful form. The underlying text that produced each assertion isn't discarded, though, because it's valuable information that keeps the knowledge in ConceptNet connected to natural language.

Each assertion in ConceptNet is connected to an alternate representation, a *raw assertion* that connects it to natural language. In the raw-assertion form, the natural-language phrases are left in their *surface form*, instead of being normalized into concepts. The edges, instead of being labeled with generalized relations, are labeled with the *frame* that the surface forms fit into to form the sentence—such as "{1} can be used to {2}."

Using the raw-assertion representation, a new assertion that's added to

ConceptNet can be represented as an understandable sentence in natural language, even if its natural-language representation isn't previously known. The raw forms of similar statements serve as a guide for the system to produce a reasonable guess at a sentence representing the new assertion.

When the OMCS Web site asks questions based on cumulative analogies, this serves to make the database's knowledge more strongly connected, because it eliminates gaps where simply no one had thought to say a certain fact. It also helps to confirm to contributors that the system is understanding and learning from the data they provide.

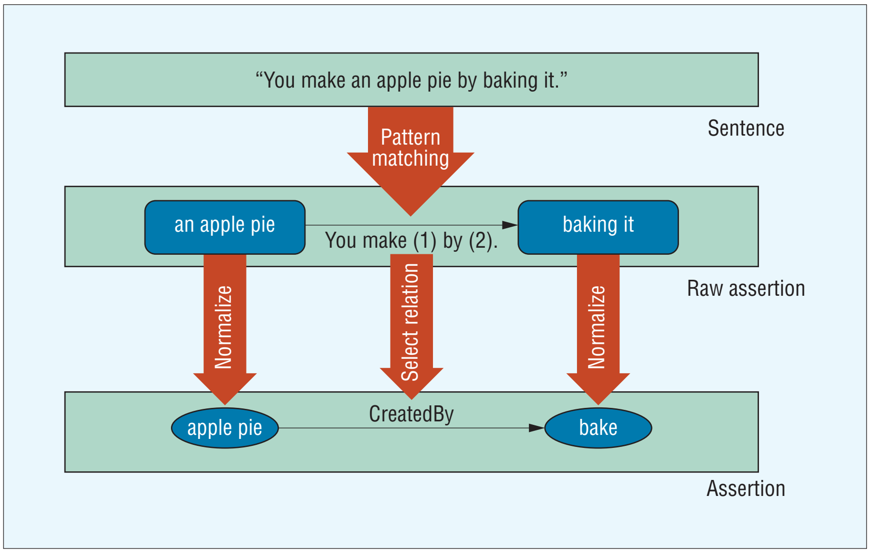

Figure 2. Adding an edge to ConceptNet based on a sentence. The sentence is transformed into a raw assertion with pattern matching, and then normalized into a ConceptNet assertion.

AnalogySpace

Any system that is to learn the things humans learn must not just survive in a noisy environment, in the presence of misinformation and imprecision; it must thrive there. Thus, a system for reasoning over a large commonsense knowledge base must be prepared to deal gracefully with inconsistency, subjectivity, and generally noisy data. Although the OMCS database contains a more than reasonable amount of correct information, it also has a noticeable amount of noise. If a reasoning system over OMCS depended on its input being fully consistent and correct, it would fail. Instead of such a reasoning system, it's important to use a method for making rough conclusions based on similarities and tendencies, not based on an assumption of absolute truth.

This goal of robustness in the presence of noise led us to develop the AnalogySpace process.[5] In this process, we represent the knowledge in a semantic network as a sparse matrix, and use singular value decomposition (SVD) to reduce its dimensionality, capturing the most important correlations in that knowledge. This generalizes a method of commonsense inference called *cumulative analogy*, first presented in Chklovski's Learner.[6]

Using SVD, any matrix **A** can be factored into an orthonormal matrix **U**, a diagonal matrix **Σ**, and an orthonormal matrix $\mathbf{V}^T$, so that $\mathbf{A} = \mathbf{U\Sigma V}^T$. The singular values in **Σ** can be ordered from largest to smallest, where the larger values correspond to the vectors in **U** and **V** that are more significant components of the initial **A** matrix. The largest singular values, and their corresponding rows of **U** and columns of **V**, represent the principal components of the data.

When making use of the SVD results, we often discard all but the first *k* components—the principal components of **A**—resulting in the smaller matrices $\mathbf{U}_k$, $\mathbf{\Sigma}_k$, and $\mathbf{V}_k^T$. The components that are discarded represent relatively small variations in the data, and the principal components form a low-rank approximation of the original data. This is called a truncated SVD, represented by this approximation:

$$\mathbf{A} \approx \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T = \mathbf{A}_k$$

We know from the properties of SVD that the result of the approximation, $\mathbf{A}_k$, is the nearest least-squares approximation to **A** of rank *k*.

This factorization allows the row space of **A** and the column space of **A** to be projected into a common space by the transformations **U** and **V**. We can think of these spaces as containing two types of objects, which we can represent as row and column vectors of **A**, which are related to each other by the values where they meet. After the SVD transformation, AnalogySpace represents both kinds of objects in the same space, where it can compare them to one another as *k*-dimensional vectors.

Prediction Using SVD

The key to discovering new information using SVD is the approximation matrix $\mathbf{A}_k$. This dense matrix contains an approximated version of the input data, expressing all the data that can be described by the first *k* principal components. Choosing the value of *k* is a practical trade-off between precision and efficiency; we tend to use *k* = 100, although there are over 400 axes that represent statistically significant amounts of variance in the data.

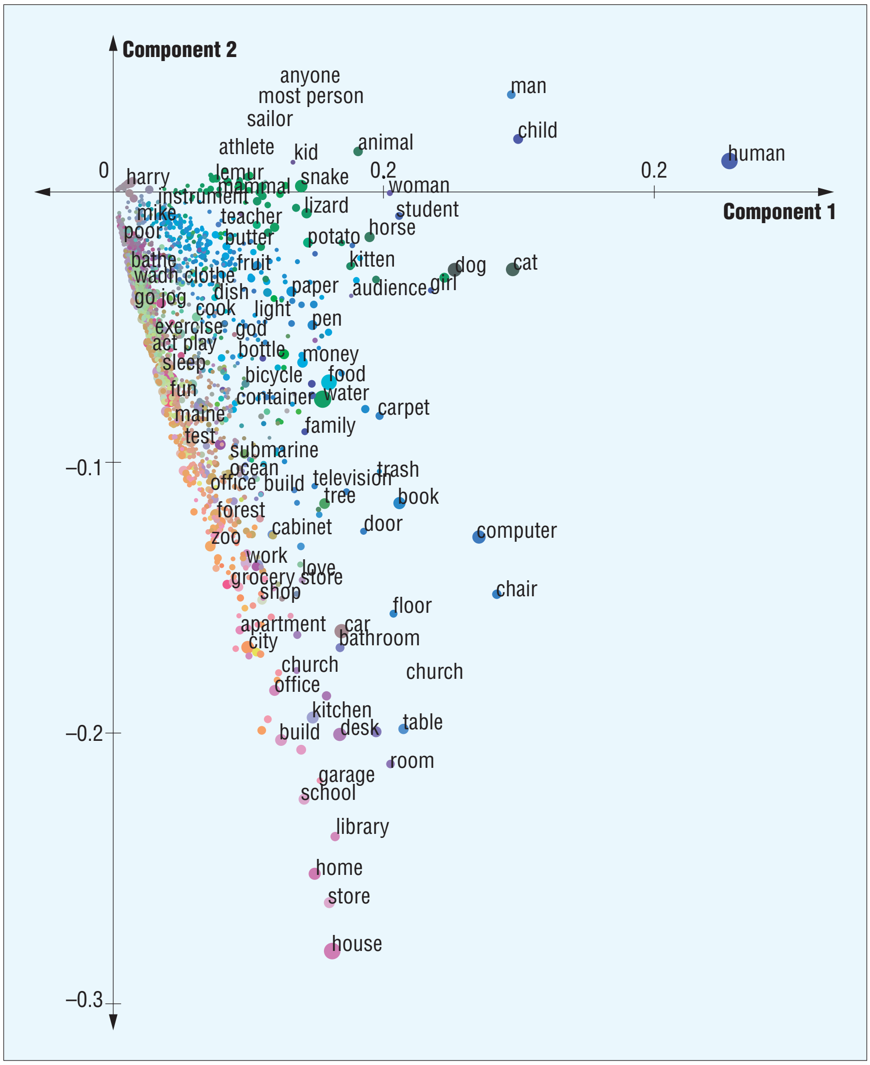

Figure 3. A two-dimensional projection of AnalogySpace. The axes are the first two principal components identified by the SVD; their signs are arbitrary. In this visualization, colors represent the values of three more components that are not shown spatially.

It's often computationally expensive to calculate $\mathbf{A}_k$ itself, as it's a dense matrix shaped like the original sparse matrix. When using only the entries in ConceptNet which have a score of 2 or greater, this dense matrix has over 2.4 billion entries. However, we know that individual entries of $\mathbf{A}_k$ result from the product of a row of U and a column of $\mathbf{V}^T$, weighted by the diagonal $\mathbf{\Sigma}$. Rows of $\mathbf{U}$ that contain vectors pointing in similar directions lead to similar results in $\mathbf{A}_k$, and the same is true for $\mathbf{V}^T$. The effect of the SVD, then, is to compress the data by sharing information between items that are similar to each other.

Applying SVD to Common Sense

To use this technique to make inferences about common sense, we express ConceptNet as a matrix. Because each assertion in ConceptNet expresses a relation between two concepts, we can describe it with the 3-tuple (*concept, relation, concept*). To fit this data into a matrix, we break each assertion into two parts, a concept and a *feature*. A feature is simply an assertion with one concept left unspecified; it expresses a feature that things may have, such as "is enjoyable" or "is part of a car."

Because either of two concepts can be omitted from an assertion, features come in two mirror forms. A *left feature*, such as "A wheel is part of . . . " contains the first concept while omitting the second; a *right feature*, such as ". . . is a kind of liquid," contains the second concept and omits the first. A given assertion can be decomposed into either a concept and a left feature, or a concept and a right feature.

The entries in the matrix defined by a concept and a feature are positive or negative numbers, depending on whether people have made positive or negative assertions that connect that concept and that feature. The magnitude of each value increases logarithmically with the confidence score. If no information is known connecting a concept to a feature, that entry of the matrix has a value of zero. Because most of the entries are unknown and therefore zero, the matrix can be represented sparsely in memory, with the zeroes implied.

The result of SVD on this matrix projects both concepts and features into the same space, where they're described as linear combinations of principal components. Figure 3 is a two-dimensional projection of this space, plotting the concepts of ConceptNet on coordinates that represent their correlation with the first two principal components. Notice that concepts cluster together in the same direction when they have similar meanings or share similar features.

The OMCS Web site uses these

predictions as a form of feedback. The site uses AnalogySpace to ask questions that seem to fill gaps in its knowledge, such as, "Would you find a park in a city?" The questions can also be generalized into fill-in-the-blank questions, such as, "What is paprika used for?" By asking the right questions, OMCS both collects valuable new data and assures users that it's learning.

This space has greatly simplified the task of finding similarities in ConceptNet. The information-sharing property of truncated SVD causes it to describe other features that could apply to known concepts by analogy: If a given concept has an unknown value for a given feature, but many similar concepts have that feature, then by analogy the given concept is likely to have that feature as well. More succinctly, concepts and features that point in similar directions (and therefore have high dot products) are good candidates for analogies. AnalogySpace uses these analogies to infer additional pieces of commonsense knowledge that aren't yet part of the database. We ask users to verify these new inferences, and they're added to the database upon verification.[5]

This technique forms a sort of middle ground between traditional symbolic and statistical techniques. As a form of statistical machine learning, AnalogySpace benefits from large amounts of data and the broad conclusions that can emerge from that data as a whole. But also, by maintaining a symbolic graph representation, AnalogySpace can explain these large-scale conclusions using small-scale examples.

Because concepts and features are projected into the same space, small groups of concepts and features that fall near each other can give meaning to an area of AnalogySpace. Focus on the concepts and features in any local area of AnalogySpace and explanations become apparent: "Offices, schools, and libraries are similar because they are all places, you might read things there, and you might go to any of them to do work."

Blending

We've seen how AnalogySpace uses SVD-based techniques to reason with and expand our commonsense knowledge. However, the power of a commonsense reasoning system isn't simply to create more common sense or to reason with the existing commonsense knowledge, but to integrate common sense into other systems. The ability to work with external data is essential to make common sense useful for something besides feeding back into itself.

The ability to work with external data is essential to make common sense useful for something besides feeding back into itself.

Earlier, we discussed using SVD methods to predict the value of entries that were unspecified in the original source matrix. We can think of the matrix of predictions that this creates as the *analogical closure* of that data—that is, the data that results when all possible analogies are made, up to a level of detail determined by the parameter *k*.

When we find an analogical closure across multiple, previously separate sources of data, we call it *blending*. Blending combines two sparse matrices linearly into a single, larger matrix that AnalogySpace can analyze with SVD.

When we perform SVD on a blended matrix, the result is that new connections are made in each source matrix taking into account information and connections present in the other matrix, originating from the information that overlaps. By this method, we can combine different sources of general knowledge, or overlay general knowledge with domain-specific knowledge, such as medical, geological, or financial knowledge.

Because ConceptNet ties its knowledge representation to natural language, blending with ConceptNet can be especially useful for work with data that's provided as natural-language text, such as tags that organize Web content, free-text responses to survey questions, or databases containing some natural-language entries that could be analyzed more deeply. Figure 4, for example, shows a blend in the domain of Boston-area businesses, given a database containing lists of products and services that they sell.

Alignment

The first step to creating a blend is to transform the input data so that it can all be represented in the same matrix. If we consider the source matrices to have sets of concepts and features that respectively index their rows and columns, the blended matrix will have the union of all the input concepts for its rows and the union of all the features for its columns.

If these concept and feature sets are disjoint, the result will be uninteresting. No analogies can be formed that cross domains unless there are either some concepts or features in common between those domains, so

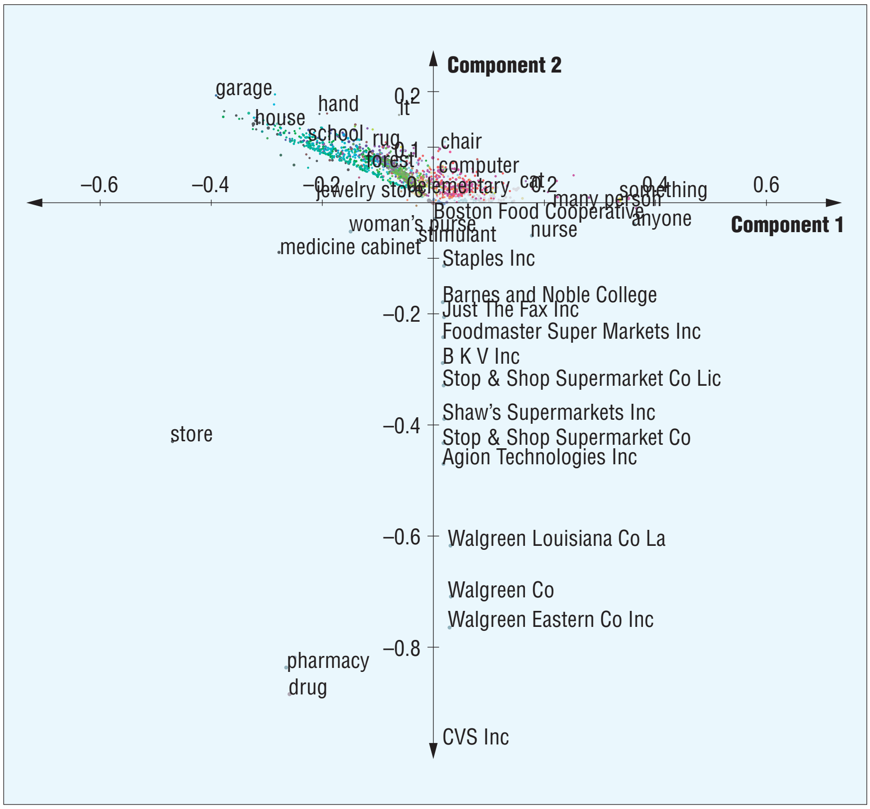

Figure 4. A projection, similar to Figure 3, of a blend between ConceptNet's general knowledge and domain-specific data on Boston-area corporations and what they sell. Points that are near each other in this space are related by common sense and domain-specific knowledge.

Table 2. WordNet relations that map onto ConceptNet relations.

WordNet relation	ConceptNet relation
Hypernym	IsA
Part holonym	PartOf
Substance meronym	MadeOf
Attribute	HasProperty
Entailment	Causes

that some rows or columns of the new matrix contain data from both sources. Alignment is the process of transforming the input data so that the different domains are described using some of the same concepts or some of the same features, which involves converting the data into some common representation. We can illustrate this with a blend of ConceptNet and the popular natural-language processing resource WordNet.[7]

Concept-based alignment is generally the more straightforward form of alignment. ConceptNet's concepts can be represented by their lemma form, with stop words and inflections removed. WordNet's *synsets* also contain words in lemma form, and rarely contain multiple-word phrases with stop words. A valid way to align ConceptNet and WordNet, then, is to align the lemma forms of ConceptNet concepts with the lemma forms of the words in WordNet (without distinguishing word senses). Whenever a ConceptNet concept and a WordNet entry have the same lemma form, there will be a row of the blended matrix that contains information from both ConceptNet and WordNet.

Feature-based alignment sometimes takes more thought. In WordNet, we identify five WordNet relations which we consider to map onto ConceptNet relations. Table 2 lists these correspondences. After aligning the concept part of the feature we've described, we can now use this table to also align the relations. The resulting aligned features can describe both ConceptNet concepts and WordNet entries.

Although in this case we have potential overlaps between ConceptNet and WordNet in both the concepts and the features, this isn't a necessity. Blending only requires overlapping data in either the concepts or the features.

In cases in which the inputs are naturally disjoint, we can create overlap by adding a third bridging data set that overlaps with both of the data sets. For example, there are no concepts or features that appear in both the English and Portuguese ConceptNets, because concepts are distinguished by language; however, we can blend the English and Portuguese ConceptNets by bridging them with assertions generated from an English-Portuguese translation lexicon.

Calculating the Blending Factor

For many blending factors, the principal components will be defined almost entirely by one matrix alone, with very little contributed by the other matrix. It's important to choose a blending factor that allows both sources to contribute to the inference.

A brief example shows how we can determine when principal components are interacting with each other. Suppose we're blending two matrices with no overlap at all. If we plot their singular values as *f* ranges from 0 to 1, we'll see these singular values decrease linearly with *f*, from their original values on one edge of the graph to 0 on the edge of the graph where that input is completely absent. The other singular values, decreasing linearly in the opposite direction, won't affect them in any way. The singular values form intersecting straight lines, as shown in Figure 5.

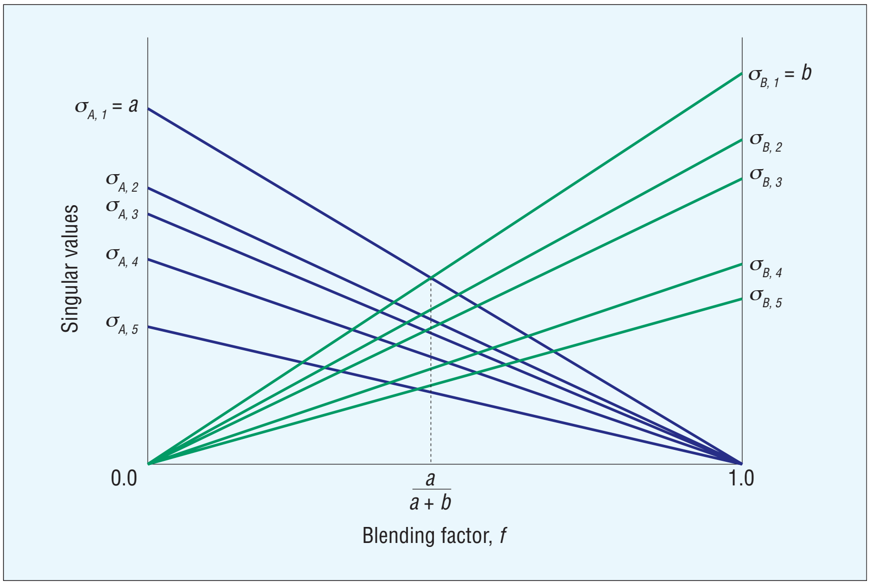

Figure 5. The singular values of a hypothetical blend of two matrices with no overlap, as blending factor *f* ranges from 0 to 1. In this case, the trajectories of the singular values intersect without interacting.

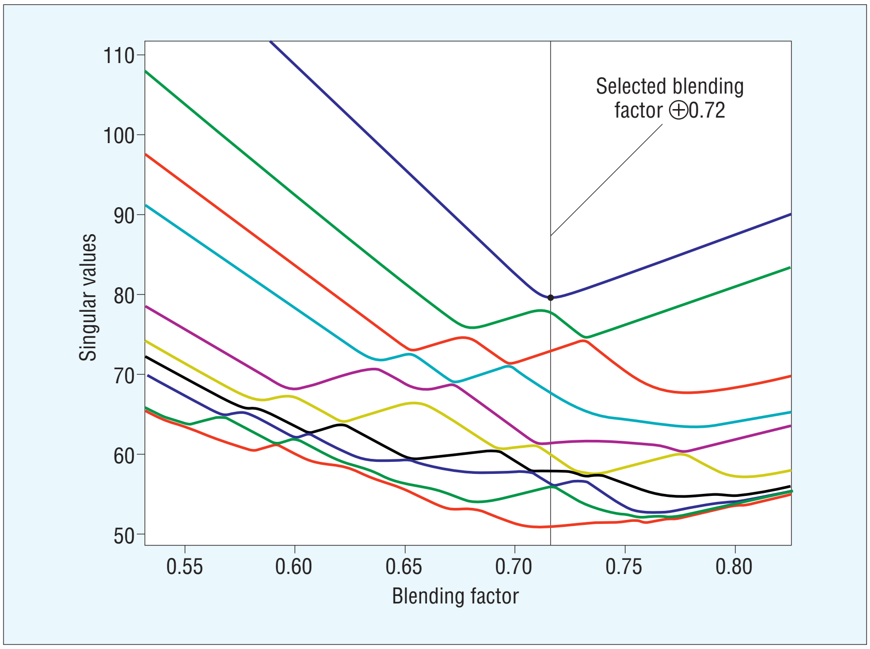

Figure 6. When two data sets contain overlapping information, blending them can produce inferences that would not be produced from either input alone. This appears as nonlinear interaction, or "veering," on a graph of the trajectories of singular values. This graph shows veering in the blend between ConceptNet and WordNet.

When there's overlap, however, the singular values don't cross each other in straight lines. Instead, some nonlinear interaction occurs around the place where they would intersect. We refer to this nonlinear behavior as *veering*. Figure 6 shows an actual example, plotting the top 10 singular values for varying values of *f* in the blend of ConceptNet and WordNet. The curved sections are the result of veering.

We theorize that veering represents the influence of analogies that span both inputs. Therefore, we want to choose the blending factor to maximize the amount of veering.

Calculating Veering

We can calculate the amount of veering that occurs by hypothesizing a blend with no interaction—one where the singular values all scale linearly in *f*—and finding the magnitudes of the differences between those hypothetical singular values and the actual ones. The time-consuming way to maximize blending is to search a range of blending factors for the one that maximizes, for example, the sum of squares of these differences. However, a quick rule of thumb gives almost the same result.

Because veering occurs around the places where singular values would otherwise meet, we can seek out a particular intersection; and a prominent one to look for is the intersection that would occur between the top singular value on each side. If the maximum singular value for one matrix is *a*, and the maximum singular value for the other matrix is *b*, these values intersect when $f = a/(a + b)$, and this represents a reasonable blending factor that tends to have a sufficiently large amount of veering.

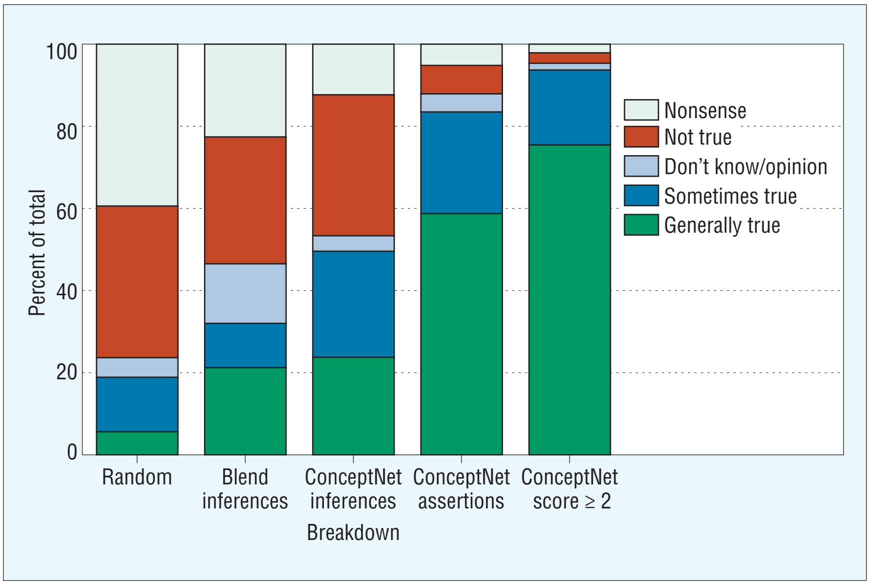

Figure 7. Open Mind Common Sense user evaluation results, January 2009. Users rated statements contributed by users to ConceptNet, predictions emerging from ConceptNet through the AnalogySpace process, sampled predictions from the blend of WordNet and ConceptNet, and random statements.

Case Study: Blending WordNet and ConceptNet

As a specific example of blending, we created a matrix of knowledge that blends together ConceptNet and a large portion of WordNet. WordNet expresses relations between synsets that contain many words. To align these with ConceptNet, for each relation between synsets we enumerated all the words it could connect, creating relations between each possible pair of words in the two synsets.

Because WordNet is intended to be a rigid hierarchy of knowledge, analogies over knowledge that comes solely from WordNet are often unhelpful or incorrect—if such an analogy produced correct knowledge, it would indicate that WordNet's creators should have stored the knowledge higher in the hierarchy in the first place. To ensure that the analogies involve at least some knowledge from ConceptNet, we filtered out all WordNet links that connected two terms that don't appear in ConceptNet, such as, "An oscine is a passeriform bird."

We used the previously described method to automatically calculate the blending factor, giving a result of about 0.72 for WordNet. Figure 6 shows this as approximately the minimum of the top line, occurring in an area where veering affects many singular values.

Evaluating Knowledge and Predictions

In keeping with the theme of human-centered knowledge collection, we use human volunteers to evaluate the quality of assertions and inferences in Open Mind Common Sense. From time to time, we ask volunteers to rate the truth and sensibility of statements from various sources.

In a 2002 study,[8] human judges evaluated the quality of a random sample of the assertions that had been collected by OMCS. They judged 75 percent of the assertions to be "largely true," 85 percent of the assertions to "make sense," 82 percent of the assertions to be "neutral," and 84 percent "common enough to be known by someone by high school."

In an evaluation process that we began in 2007,[5] volunteers assign ratings to a random sampling of statements selected from various stages of the OMCS system, such as user-verified assertions in ConceptNet and predictions made by AnalogySpace, all expressed in natural language using the same process. As a control, some of the statements are random nonsense assertions constructed by combining randomly selected concepts and features. Each user sees the same number of assertions from each source, mixed together in a random order so the evaluation is blind and impartial.

For each assertion, users choose a rating from the set "generally true," "sometimes true," "not true," "doesn't make sense," and "not true but amusing." As of the 2009 evaluation, we collect the "not true but amusing" assertions and show some to each participant as a small reward for their time. We count these assertions as "not true" when graphing the results.

In the most recent evaluation, with 59 participants rating 60 statements each, the statements came from four sources. One quarter are existing assertions that users have contributed to ConceptNet; those with a score of at least 2 (indicating that they have been confirmed by at least one other user) are distinguished from the others. One quarter are predictions that emerge from ConceptNet using the AnalogySpace process, sampled with a probability proportional to their predicted value. Another quarter are sampled predictions from our example blend, incorporating information from WordNet and ConceptNet. The final quarter are random. Figure 7 reports the results.

We evaluate our inference processes with respect to multiple goals. Of course, it's always beneficial to be able to automatically infer correct new knowledge, so a system that makes use of common sense can fall back on reasonable assumptions in

Table 3. Mean scores of random statements, ConceptNet inferences, and inferences from ConceptNet blended with WordNet.

Source	Score (99% confidence)*
Random	1.668 (± 0.205)
Blended inferences	2.494 (± 0.152)
ConceptNet inferences	2.939 (± 0.194)

* A score of 0 corresponds to "doesn't make sense"; 5 corresponds to "generally true."

the absence of the correct knowledge. That establishes a goal of making as many inferences as possible that are "generally true" or "sometimes true." In the presence of noisy data and ambiguity, though, some inferences will inevitably be incorrect. We've found that users have greater confidence in a system, whether it's OMCS itself or a derived application, if the errors it makes are at least plausible. This establishes a secondary goal of avoiding "nonsense" results. Finally, we want inference processes to expand the domain of the data, making inferences that wouldn't normally appear in the corpus; this is the motivation for blending different resources together.

To compare the correctness of the predicted statements, we translated the responses into an integer scale, giving a score of 0 for "doesn't make sense," 1 for "not true but amusing," 2 for "not true," 3 for "don't know" or "opinion," 4 for "sometimes true," and 5 for "generally true." We measured the mean rating each participant gave to each source of assertions. By comparing the scores of both kinds of inferences to the random statements using a paired t-test, we can show that the inferences were better than random assertions, both when they are produced from ConceptNet alone ($t = 10.61$, $df = 58$, $p > .99$) and from the blended data ($t = 13.49$, $df = 58$, $p > .99$). Table 3 lists the mean scores for the different kinds of inferences, as well as for the random baseline.

As expected, the statements most often considered truthful by participants were the ones that were already present in ConceptNet, particularly the subset of those that had been confirmed (of which 94 percent were rated "sometimes true" or better). Predicted assertions were rated significantly higher than random assertions, with more true statements and less nonsense.

The predictions that emerged from the blend of WordNet and ConceptNet expanded the vocabulary beyond the concepts in ConceptNet, including inferences such as, "A workday is a part of February," and "A synchroflash is a machine." This came at the expense of some accuracy: the blend created considerably more nonsense inferences. Also, because some of these inferences fell outside the domain of common knowledge, they showed a significant increase in the number of "don't know/opinion" results. These results were still significantly better than the random ones, and could be used as a basis for a system that would ask questions based on WordNet to expand ConceptNet's vocabulary.

We can use a process much like the WordNet blend we've described to aid in bootstrapping new lexical resources. Creating linguistic resources of any kind is a time-consuming, expensive process. It often takes years and many annotators to create such a data set, and funding for these projects is often hard to find. A process that can automatically suggest connections to add to an incomplete resource can save time, making the obvious parts easier to add while allowing the experts to focus on imparting their expert knowledge.

Blending doesn't require all of its input to be structured knowledge—with the right form of alignment, it can work with other modalities of common sense. We've previously mentioned using the data in ConceptNet to better understand a natural-language resource. One way to work with natural language is to use dependency parsing to make graphs of a free-text resource, and to make a blend that includes a representation of those graphs and seed data that represents the semantics of some text. In particular, we're interested in doing this with ConceptNet and the OMCS corpus itself—this will allow us to extract more information from the contributed text that we haven't yet been able to parse.

We've also explored using this approach in knowledge management, helping to construct and make use of databases of domain-specific, task-specific, and expert knowledge. Working with partner corporations, we've developed a method for creating specific knowledge bases with information about a topic, such as geoscience or finance, collecting knowledge from games, specific knowledge entry, and targeted acquisition from other resources both structured and unstructured. Organizations can use these blended knowledge bases to help organize and understand documents and other data, visualize and simplify trends in data and opinions, and understand customers' opinions and goals.

A step beyond this would be to construct a blend out of cross-modal data, to incorporate into our commonsense system other forms of grounded knowledge such as images or sound samples with descriptions.

Other Collections of Common Sense

In addition to Open Mind Common Sense, other projects have undertaken the task of collecting commonsense knowledge. The Cyc project,[1] started by Doug Lenat in 1984, aims to represent human knowledge as a set of interconnected knowledge representations with a formal logical language at the core. Cyc, along with other resources such as WordNet, FrameNet, PropBank, and the Brandeis Semantic Ontology (BSO), collects its knowledge from trained knowledge engineers.

Lenhart Schubert's Epilog system[2] is a different logically driven approach to common sense acquisition, using a representation based on episodic logic. This long-running project was first released in 1990. Epilog has the ability to extract general relations from corpora of free text. Epilog does not, however, have a commonsense knowledge base as a primary part of its project.

Interestingly, ConceptNet, Cyc, and WordNet all contain a similar number of nodes overall. As of March 2009, Cyc contains "nearly two hundred thousand terms,"[3] WordNet has 206,941 synsets,[4] and ConceptNet contains 204,279 concepts.[5]

References

1. D. Lenat, "CYC: A Large-Scale Investment in Knowledge Infrastructure," *Comm. ACM,* vol. 38, no. 11, 1995, pp. 33–38.
2. L.K. Schubert, "Can We Derive General World Knowledge from Texts?" Proc. *Human Language Technology Conf.,* Morgan Kaufmann, 2002, pp. 24–27.
3. "What's in Cyc?"; http://www.cyc.com/cyc/technology/whatiscyc_dir/whatsincyc, accessed March 2009.
4. "WordNet 3.0 Database Statistics"; http://wordnet.princeton.edu/man/wnstats.7WN, accessed March 2009.
5. Retrieved from the ConceptNet 3.5 database, March 2009.

If we give a commonsense system access to audio and visual data that's partially aligned with linguistic data, it would have access to a wider range of representations, which it could use to solve more problems and get input from an even broader portion of the real world.

Many problems require common sense to solve, and different problems benefit from having different data available to them. Common sense can be the source for the intuitive links and connections that form the underlying base of our reasoning about the world, no matter what sits on top of that base. Using SVD to analyze a graph representation, and using blending to expand the range of this representation, we can provide a system with digital intuition that helps it to relate to the world and its inhabitants. ■

Acknowledgments

The Common Sense Computing Initiative thanks the tens of thousands of users who have entered, checked, and rated our knowledge, as well as our multilingual collaborators at other universities. We also thank Jason Alonso, Kenneth Arnold, Jayant Krishnamurthy, and our undergraduate researchers for their help developing and using the AnalogySpace process. For their support and interaction, we thank our sponsors, especially Microsoft, Schlumberger, and many collaborators from Bank of America through the Center for Future Banking at the MIT Media Laboratory.

References

1. P. Grice, "Logic and Conversation," *Speech Acts,* Academic Press, 1975.
2. C. Havasi, R. Speer, and J. Alonso, "ConceptNet 3: A Flexible, Multilingual Semantic Network for Common

THE AUTHORS

Catherine Havasi is one of the cofounders of the Open Mind Common Sense project at the Massachusetts Institute of Technology (MIT) Media Lab. This is her last year as a PhD student in computer science at Brandeis University with James Pustejovsky, and her interests include generative lexicon, commonsense reasoning, dimensionality reduction, machine learning, language acquisition, cognitive modeling, and intelligent user interfaces. She has an SB in computer science and an MEng for cognitive modeling work, both from MIT. Contact her at havasi@cs.brandeis.edu.

Robert Speer is a graduate student at the Massachusetts Institute of Technology doing research in the Media Lab's Commonsense Computing Initiative. His interests include natural-language processing, machine learning, and multilingual commonsense reasoning. He has an SB in computer science, an SB in music, and an MEng from MIT. Contact him at rspeer@mit.edu.

James Pustejovsky is a professor of computer science at Brandeis University, where he runs the Laboratory for Linguistics and Computation. He has developed a theory of lexical semantics known as the Generative Lexicon, and is the author of several books, including The Generative Lexicon and Lexical Semantics and the Problem of Polysemy. His research interests include computational linguistics, lexical semantics, temporal reasoning and annotation, discourse reasoning, and temporal and spatial ontologies. He has a PhD from the University of Massachusetts, Amherst. Contact him at jamesp@cs.brandeis.edu.

Henry Lieberman is a research scientist at the Massachusetts Institute of Technology (MIT), where he directs the Software Agents group, which is concerned with making intelligent software that provides assistance to users in interactive interfaces. He co-edited the books End-User Development and Spinning the Semantic Web and edited Your Wish is My Command: Programming by Example. He holds a doctoral-equivalent degree (Habilitation) from the University of Paris VI. Contact him at lieber@media.mit.edu.

Sense Knowledge," *Recent Advances in Natural Language Processing* (RANLP 07), John Benjamins, 2007; http://web.mit.edu/~rspeer/www/research/cnet3.pdf.

3. H. Liu and P. Singh, "ConceptNet: A Practical CommonSense Reasoning Toolkit," *BT Technology Journal*, vol. 22, no. 4, 2004, pp. 211–226.
4. A. van den Bosch and W. Daelemans, "Memory-Based Morphological Analysis," *Proc. 37th Ann. Meeting Assoc. for Computational Linguistics*, ACM Press, 1999, pp. 285–292.
5. R. Speer, C. Havasi, and H. Lieberman, "AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge," *Proc. AAAI Conf. Artificial Intelligence* (AAAI 08), pp. 548–553, AAAI Press, 2008.
6. T. Chklovski, "Learner: A System for Acquiring Commonsense Knowledge by Analogy," *Proc. 2nd Int'l Conf. Knowledge Capture* (K-CAP 03), ACM Press, 2003, pp. 4–12.
7. C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.
8. P. Singh et al., "Open Mind Common Sense: Knowledge Acquisition from the General Public," *On the Move to Meaningful Internet Systems,* Lecture Notes in Computer Science, vol. 2519, Springer-Verlag, 2002, pp. 1223–1237.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.