AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge

Robert Speer
CSAIL
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
[redacted]

Catherine Havasi
Laboratory for Linguistics and Computation
Brandeis University
Waltham, MA 02454, USA
[redacted]

Henry Lieberman
Software Agents Group
MIT Media Lab
Cambridge, MA 02139, USA
[redacted]

Abstract

We are interested in the problem of reasoning over very large common sense knowledge bases. When such a knowledge base contains noisy and subjective data, it is important to have a method for making rough conclusions based on similarities and tendencies, rather than absolute truth. We present AnalogySpace, which accomplishes this by forming the analogical closure of a semantic network through dimensionality reduction. It self-organizes concepts around dimensions that can be seen as making distinctions such as "good vs. bad" or "easy vs. hard", and generalizes its knowledge by judging where concepts lie along these dimensions. An evaluation demonstrates that users often agree with the predicted knowledge, and that its accuracy is an improvement over previous techniques.

Introduction

This paper introduces AnalogySpace, a new technique designed to facilitate reasoning over a large knowledge base of natural language assertions that represent common sense knowledge. Reasoning about common sense poses some unique challenges. Unlike traditional mathematical logic, common sense knowledge is often imprecise and inconsistent, causing problems for traditional proof procedures. We are often less interested in determining the absolute truth of a factual proposition, as we are in computing somewhat intangible qualities such as context, connotations, tendencies, analogies, and patterns of similar relationships.

AnalogySpace represents a new synthesis between symbolic reasoning techniques and statistical methods. Like symbolic reasoning, the source material for our reasoning is a set of assertions, which in our case are expressed in natural language or a derivative representation, rather than in logic. AnalogySpace learns from this source data using a technique similar to the information retrieval technique of latent semantic analysis (LSA), because of its ability to construct large-scale patterns out of a myriad small of bits of evidence. Unlike traditional LSA, however, AnalogySpace works with semantically strong assertions, rather than the weak semantics of word co-occurrence in documents.

AnalogySpace, using data from the Open Mind Common Sense (OMCS) project, represents knowledge as a matrix of objects or concepts along one axis, and features of those objects along another, yielding a sparse matrix of very high dimension. We then use singular value decomposition (SVD) to reduce the dimensionality of that matrix. This results in computing *principal components* which represent the most salient aspects of the knowledge, which can then be used to organize it along the most semantically meaningful dimensions. The key idea is that semantic similarity can be determined using linear operations over the resulting vectors.

What AnalogySpace Can Do

AnalogySpace provides a computationally efficient way to calculate a wide variety of semantically meaningful operations:

AnalogySpace can generalize from sparsely-collected knowledge. The amount of common sense knowledge needed in everyday life is extraordinary: estimates say that we have at least several million facts (Chklovski 2003) at our disposal. It has been theorized that much of semantic meaning is generative (Pustejovsky 1998), and thus we believe that people are likely to generalize some of their common sense information by creating an analogy to existing knowledge. It is important, then, for a knowledge base to be able to be generalized in the same way, and dimensionality reduction gives us a way to do so by inferring new features for concepts in a knowledge base.

AnalogySpace can classify information in a knowledge base in a variety of ways. The reduced-dimensionality space that results from principal components analysis (PCA) is the one that best describes the variations in the knowledge. Different vectors in this space represent different ways of making binary distinctions among the concepts in ConceptNet, by projecting all of the concepts onto one line.

Interesting distinctions come from vectors that align with a meaningful cluster of concepts, and some of them are suggested by the principal components themselves. For example, we can observe that the most significant component of AnalogySpace distinguishes things people want from things people do not want. Concepts that fall in the "people want" direction include *love*, *money*, and *vacation*, while the opposite direction includes *lose your keys* and *slavery*. By projecting concepts onto other vectors, we can make other distinctions such as "urban vs. rural", "animate vs. inanimate", and "indoors vs. outdoors".

AnalogySpace can create "ad hoc categories" and provide common sense justifications for why things belong to those categories. Our notion of generalized similarity makes it easy to extend a category using a few examples. For example, typing "knife, fork, spoon" into our categorizer will result in a category that also includes "plate" and "chopsticks".

EFTA01154696

AnalogySpace can also list the common sense features that justify the classification, such as “found in the kitchen” and “used to eat food”.

AnalogySpace can confirm or question existing knowledge. AnalogySpace can assign an inference score to any assertion, including assertions that already exist in the database. This determines which assertions are well supported by other knowledge, and which appear dubious and should be checked by humans. This procedure has been used to identify bad assertions in ConceptNet such as “Books taste good”.

Common Sense Computing

To improve computers’ understanding of the world and the way they interact with users, we must teach them about the concepts and relationships which underlie everything we know and talk about. Common sense knowledge is the basis of this network of information, expressing the relations between concepts that we take for granted. However, this knowledge can be difficult to capture, because much of it is considered so obvious that it is rarely mentioned in corpora.

The Open Mind Common Sense Project

In 2000, what would become the Common Sense Computing Initiative started the Open Mind Common Sense (OMCS) project, asking volunteers on the internet to enter common sense information through a website (Singh *et al.* 2002). OMCS has collected over 700,000 pieces of common sense information in English alone from 15,000 contributors, and has expanded to several other languages.

ConceptNet

Many kinds of statements that users have entered into Open Mind can be expressed as relationships between two concepts, which are essentially short phrases of natural language. This fact was used to create ConceptNet, the semantic network representation of the OMCS knowledge base (Havasi, Speer, & Alonso 2007; Liu & Singh 2004). ConceptNet is made up of a network of *concepts* connected by one of about twenty *relations* such as “IsA”, “PartOf”, or “UsedFor”, which are labeled as expressing positive or negative information using a *polarity* flag. The relations are based on the most common types of knowledge entered into the OMCS database, both through free text entry and semi-structured entry. For the assertion “A trunk is part of a car”, for instance, the two concepts are “trunk” and “car”, the relation is “PartOf”, and the polarity is positive. For the assertion “People don’t want to be hurt”, the concepts are “person” and “hurt”, the relation is “Desires”, and the polarity is negative.

The current version of ConceptNet contains over 250,000 assertions. 3.4% of these assertions have negative polarity.

ConceptNet is useful because it provides applications with a connection between natural language text and an understanding of the world (Lieberman *et al.* 2004). A specific example of its use is to improve the accuracy of speech recognition (Lieberman *et al.* 2005) by selecting words that make sense in context.

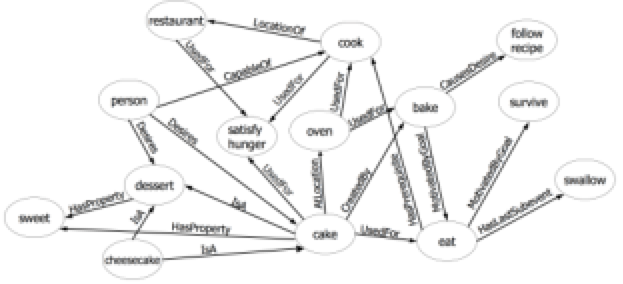

Figure 1: An illustration of a small section of ConceptNet.

Learner and Cumulative Analogy

Tim Chklovski used knowledge extracted from Open Mind to create the Learner system (Chklovski 2003). With Learner, he introduced the idea of reasoning about common sense by “cumulative analogy”, a form of inference by induction.

First, statements of common sense knowledge are divided into *objects* (analogous to ConceptNet’s *concepts*) and *features*, which are descriptions of objects that complete a statement about them, such as “is a part of a car”. The similarity between two objects is defined as the number of features they have in common. The *analogy* step, then, is to hypothesize that features that hold for one object also hold for similar objects. If many of an object’s nearest neighbors by similarity have a certain feature, this creates a strong inference that that object has the feature as well.

Let us suppose, for the sake of example, that such a system doesn’t know much about cats. To begin with, it knows that “a cat is a pet”, “a cat has fur”, and “a cat has a tail”. It knows slightly more about dogs, including “a dog is a pet”, “a dog has fur”, “a dog has a tail”, *and* “a dog has four legs”. Based on the three existing similarities in its knowledge about cats and dogs, it could transfer the “has four legs” feature from the *dog* concept to the *cat* concept, correctly inferring that “a cat has four legs”.

Cumulative analogy is the ideological predecessor to AnalogySpace, which uses a vector space of potential analogies to add more power, efficiency and resistance to noise to the process.

AnalogySpace

The Motivation for Dimensionality Reduction

In order to draw new conclusions from analogies, we need to be able to identify similarities between concepts in ConceptNet, with a technique that retains its effectiveness as the number of concepts and features increases. Concepts that share features that are themselves similar, but not identical, are missed by the basic process of cumulative analogy. This motivates us to use truncated singular value decomposition to discover similarities, which simultaneously reduces the dimensionality of our data and generalizes the notion of similarity to one that is less brittle.

The crucial observation to make is that similarity is a linear operation over vectors. This makes it possible to gen-

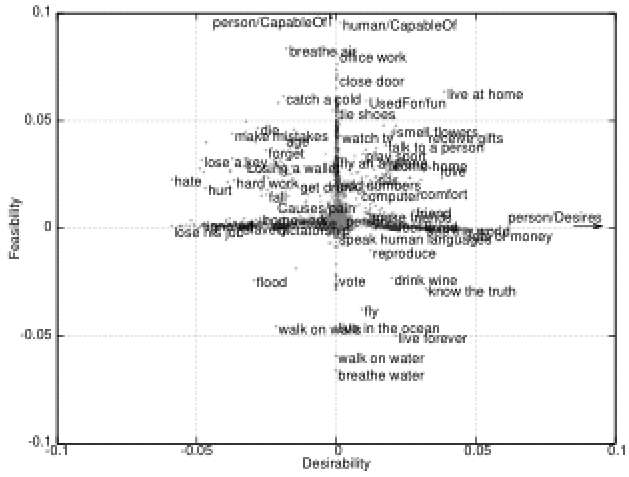

Figure 2: AnalogySpace projected onto its first two components, with some concepts and features labeled. We interpret these components as representing the notions of "desirability" and "feasibility". The features "Person/Desires" and "Person/CapableOf" have very large values on these respective axes, indicated by arrows pointing off the graph.

eralize and optimize the cumulative analogy process using singular value decomposition, the linear algebra technique used in principal component analysis and latent semantic inference.

As in Learner, we define *features* as the complement of concepts. Because assertions relate two concepts, each assertion can be decomposed into a concept and a feature in two ways, by focusing on the concept from each side. For example, the assertion which states that "a trunk is part of a car" applies the feature (PartOf, "car") to the concept "trunk", and also applies the feature ("trunk", PartOf) to the concept "car".

Each concept can then be associated with a vector in the space of possible features. The values of this vector are positive for features that produce an assertion of positive polarity when combined with that concept, negative for features that produce an assertion of negative polarity, and zero when nothing is known about the assertion formed by combining that concept with that assertion. As an example, the feature vector for "steering wheel" could have $+1$ in the position for "is part of a car", $+1$ for "is round", and -1 for "is alive". (We will address more specifically what positive and negative values we assign later, in the *Scores and Normalization* section). These vectors together form a matrix whose rows are concepts, whose columns are features, and whose values indicate truth values of assertions.

The degree of similarity between two concepts, then, is the dot product between their rows in the concept/feature matrix. The value of such a dot product increases whenever two concepts are described with the same feature, and decreases when they are described by features that are negations of each other. When performed on the concept/feature matrix, however, these dot products have very high dimensionality (as many dimensions as there are features) and are difficult to work with.

The process of truncated singular value decomposition lets us *approximate* these dot products in a useful way. A truncated SVD projects all of the concepts from the space of features into a space with many fewer dimensions. It also performs the dual operation, projecting *features* from a space of concepts into the same reduced-dimensional space. We can now run the cumulative analogy process in this new, more compact space of concepts and features, which we call AnalogySpace.

Building AnalogySpace

The principle of singular value decomposition (Wall, Rechtsteiner, & Rocha 2003) is that any matrix A can be factored into an orthonormal matrix U, a diagonal matrix Σ, and an orthonormal matrix V^T, so that $A = U\Sigma V^T$. The singular values in Σ are 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.

We discard all but the first k components — the principal components of A — resulting in the smaller matrices U_k, Σ_k, and V_k^T. The components that are discarded represent relatively small variations in the data, and the principal components form a good approximation to the original data. This is called a *truncated SVD*, representing the approximation $A \approx A_k = U_k \Sigma_k V_k^T$.

As AnalogySpace is an orthogonal transformation of the original concept and feature spaces, dot products in AnalogySpace approximate dot products in the original spaces. This fact can be used to compute similarity between concepts or between features in AnalogySpace. We call the result *generalized similarity*.

Concepts that ought to be similar, but share no exact features, get an ordinary similarity value of 0, but their generalized similarity can be a positive number. So generalized similarity allows a useful similarity value to be calculated for any two concepts, not just concepts with exact features in common.

The truncated SVD can be seen as a way of finding a space of "eigenconcepts" – k independent linear combinations of concepts that span the k dimensions of AnalogySpace – and representing both concepts and features as linear combinations of eigenconcepts. Specifically, if there are m concepts and n features, the $m \times n$ matrix representing ConceptNet is factored into:

- U_k, a $m \times k$ orthogonal matrix that relates concepts and eigenconcepts
- Σ_k, a $k \times k$ diagonal matrix of singular values, which assign a weight to each eigenconcept
- V_k, a $n \times k$ orthogonal matrix that relates features and eigenconcepts.

A plot of the first two components of AnalogySpace, representing both U_2 and V_2, is shown in Figure 2.

Making Inferences using AnalogySpace

In the AnalogySpace representation, concepts and features can be directly compared, now that they are both represented in terms of eigenconcepts. In fact, if we take the dot product of a concept and a feature, while weighting each eigenconcept by the singular value that represents how significant it is, we simply get the defining equation for the truncated SVD: $A_k = U_k \Sigma_k V_k^T$.

Each row of A_k corresponds to a concept, and contains a vector that assigns numerical values to its features. These values become larger than they were in the original matrix A when they belong to multiple similar concepts. This is exactly what cumulative analogy is meant to do. In short, A_k is not just an approximation, it is the closure of cumulative analogy over ConceptNet.

Another way to view the truncated SVD is that it "smooths" the somewhat noisy data in ConceptNet. Assertions which do not correlate well with other assertions are discarded, while large-scale similarities between many concepts are emphasized. Individual concepts are no longer associated with a complete list of their features; instead, they are blurred together somewhat with similar concepts. We take advantage of this smoothing effect to make inferences in AnalogySpace.

Scores and Normalization

Every existing assertion in ConceptNet has an integer confidence score, which is initially 1. This score is automatically increased when multiple users enter the same assertion and decreased when users enter contradictory assertions, and users of the Web site can increment or decrement the score of an assertion by 1. Assertions with zero or negative confidence are not used in the matrix.

We combine the confidence score with the polarity when assigning numerical values to assertions in the matrix. Assertions with positive polarity are assigned their confidence score as a value, while negative assertions are assigned the negation of their confidence score.

Before computing the SVD, we normalize the rows of the matrix so that a few high-information concepts such as "person" do not dominate all the others. We scale the rows down by their Euclidean norm, plus a small additional term; this makes all concepts have nearly the same magnitude, with the magnitude decreasing for very low-information concepts.

Parameters

Because it is parsed from free text, ConceptNet contains some spurious concepts that appear, for example, in only one assertion. Very rare concepts make the SVD take longer to compute for little benefit, so we set a minimum on how many assertions a concept needs to be involved in before we represent it in AnalogySpace. We currently use a minimum of 4 assertions. (In other research, we are working on processes that encourage Open Mind users to "bootstrap" new concepts, answering questions about them that provide enough information to represent them in AnalogySpace.)

As a parameter to the SVD process, we need to specify k, the size of the truncated matrices – in other words, the number of singular values or eigenconcepts to find. The sparse SVD implementation we use, Doug Rohde's `svdlibc`, can be asked to produce all singular values it considers statistically significant, in which case it produces over 400 singular values for AnalogySpace. However, we find that too large of a value of k makes inference run slower while generalizing less, so we typically use $k = 50$ or $k = 100$. A value of $k = 50$ was used for the evaluation in this paper.

Open Mind Commons

The inferences produced by AnalogySpace have an immediate purpose, completing a cycle of feedback in the way that Open Mind learns. Open Mind Commons (Speer 2007) is a knowledge collection interface that runs on top of ConceptNet 3, bridging the gap between the computer-understandable knowledge and contributors. When contributors enter new statements through this interface, they immediately become part of ConceptNet. They are then used to infer new possible assertions, which are presented to the user for approval. This helps make the knowledge entry process more interactive, and aids us by encouraging users to enter assertions whose answers will fill gaps in ConceptNet.

Related Work

Other Methods for Acquiring Common Sense

Some projects aim to extract general knowledge from existing corpora. Suh, Halpin, & Klein (2006) use named entity recognition and noun and verb group detection to extract common sense statements from Wikipedia, and (Eslick 2006) has used data mining techniques to extract common sense from websites on the Internet. The KNEXT project (Schubert 2002) uses patterns to extract semantic relationships from the Penn Treebank. The knowledge collected by these projects is of a different nature than the common sense OMCS aim to collect, because it does not include the set of fundamental knowledge that is so "obvious" that people leave it unstated.

The Cyc project (Lenat 1995) is another attempt to collect common sense knowledge. Started by Doug Lenat in 1984, this project utilizes knowledge engineers who handcraft assertions and place them in Cyc's logical frameworks, using a logical representation called CycL. Cyc has traditionally relied on trained knowledge engineers to acquire their knowledge but has recently made efforts to acquire knowledge from other sources. For example, the goal of the WAC project (Coursey 2007) is to let Cyc discover patterns in its own data using the Weka (Witten & Frank 2005) classifier. The project then uses this classifier to allow Cyc to classify whether email is spam based only on its subject line.

PCA for Semantic Relationships

The most well-known use of principal component analysis, or singular value decomposition, in language is in latent semantic analysis (LSA), which attempts to find similarities in the domains of words and documents (Deerwester *et al.* 1990). LSA is often used in information retrieval, where it is also known as latent semantic indexing. In the representation used by LSA, a document is seen as an unordered collection of words, and the matrix of words versus documents is analyzed with SVD, so that documents are sorted into implicit categories according to the words that are contained in them.

Various research projects have expanded the use of SVD in semantic contexts to examine broader contexts than words within documents. An approach used by Patwardhan & Pedersen (2006) is to replace the documents in LSA with sentences. This reduced connections made between words or concepts that are far from each other in a document.

Turney (2005) creates a matrix using the co-occuring words as the rows and the context in which the words appear within a sentence as the columns. This project aims at creating more literal analogies: its goal is to answer the kind of analogy questions that would appear on a standardized test, such as "quart:volume :: mile:distance".

The field of applying SVD to lexical resources themselves is small so far. Banerjee & Pedersen (2003) use the co-occurrence of words in WordNet glosses as a measure of semantic similarity. This measure of similarity is based only on dictionary-like glosses within WordNet, rather than the connections in WordNet itself. Perhaps the closest thing to our application of SVD is another project descended from Open Mind Common Sense, using Honda's domain-specific Open Mind Indoor Common Sense corpus. The constructed second-order SVD mapping concept to concept within an individual relation type, such as PartOf or AtLocation (Gupta & Kochenderfer 2004). However, this technique cannot learn by synthesizing information expressed with different relations. Because this technique is the closest to the methods we have used, we choose to evaluate our system in comparison to this algorithm.

Evaluation

We use AnalogySpace to fill gaps in ConceptNet's common sense knowledge and to ask relevant questions to contributors, so our goal is for it to produce inferences that make sense to users, and which they frequently consider true. The truth of inferences is important in unsupervised uses of AnalogySpace, and in the context of user interaction, it assures users that the system is learning from their input. To evaluate the success of inference using AnalogySpace, we asked users to evaluate the truth of assertions produced by AnalogySpace, as compared with assertions produced with other sources.

Our subjects were 40 college students and recent college graduates, most of whom had never interacted with Open Mind before, taking this evaluation on a special page of the Open Mind Commons web site. No compensation was offered. We presented each subject with a list of 60 assertions, converted from assertions to English sentences using a natural language procedure in Open Mind Commons.

The assertions were produced from four sources and shuffled together. 25% of them were existing assertions in ConceptNet, entered by human contributors, sampled randomly from all those with a confidence score of at least 2. Another 25% were produced by AnalogySpace but did not already exist in ConceptNet, and were sampled with a probability function that increases with their inferred score. 25% more were sampled in the same way from a modification of AnalogySpace to emulate the previous work by Gupta & Kochenderfer (2004), by using a separate SVD for each relation instead of a single combined SVD, and using the same number of principal components as AnalogySpace ($k = 50$) for each SVD. We refer to this implementation below as the "within-relations SVD". The final 25% of the assertions were nonsense, generated from random combinations of concepts and features.

Participants rated each assertion with one of the choices "Generally true", "Sometimes / Somewhat true", "Don't know / Opinion", "Generally false", "Doesn't make sense", and "Not true but amusing" (an option suggested by participants in a previous study). Figure 3 shows the breakdown of ratings that participants gave to assertions from all four sources. The performance of both methods of inference was much closer to that of existing assertions than to randomly-generated assertions. The most prominent difference between AnalogySpace and the within-relations SVD is that AnalogySpace produced considerably fewer assertions that were rated as untrue or nonsensical.

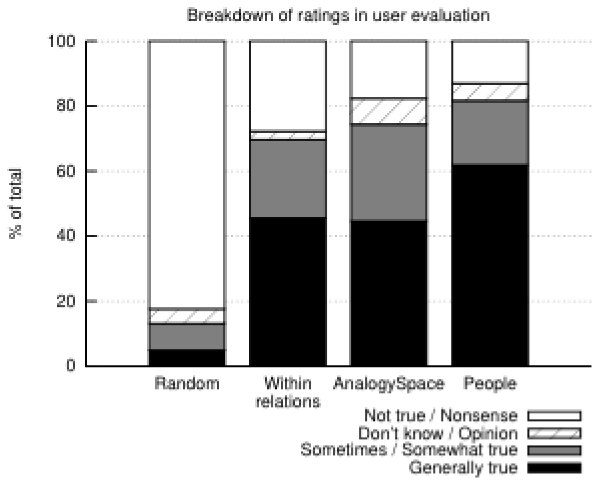

Figure 3: The ratings given by participants in the study to assertions from four different sources.

In order to quantitatively compare these ratings, we mapped the rating choices to scores, where "Generally true" was worth 2, "Sometimes / Somewhat" worth 1, "Don't know / Opinion" worth 0, and all other options were worth -1. On this scale, the existing assertions in AnalogySpace received an average score of 1.315. The new assertions inferred by AnalogySpace scored 1.025, outperforming the within-relations SVD which scored 0.882. The random assertions, as expected, scored much lower at -0.644.

Next, we recalibrated the average ratings for each subject, to take into account the fact that they would have differing interpretations of the options. We linearly mapped each user's scores to a scale where 0 represented that user's average rating of random assertions, and 100 represented that user's average rating of assertions contributed by people.

On this scale, the within-relations SVD received a mean score of 78.1 and a standard deviation of 18.5, while AnalogySpace received a mean score of 87.5 and a standard deviation of 22.2. These large standard deviations come from the fact that different users used the scale very differently, which the reason why we have calibrated the ratings per user. Comparing the calibrated scores *for each user* using a paired t-test showed that the average 9.4 point difference was statistically significant at the $p < 0.05$ level, with a 95% confidence interval of (1.0, 17.8). We conclude from this that AnalogySpace's use of features that involve many different relations gives it additional inferential power over the previous state of the art.

Discussion

By reducing the dimensionality of the sparse knowledge in ConceptNet, AnalogySpace generalizes that knowledge and makes it more broadly useful to applications. AnalogySpace's ability to predict new assertions has already helped to improve the way that OMCS collects its knowledge, by giving its users feedback in the form of relevant inferences. The fact that these inferences are frequently true or at least sensible, as shown by our user evaluation, helps to give our users confidence that the system is learning from their input.

AnalogySpace can be used to accomplish other tasks besides filling in the gaps of ConceptNet. As mentioned in the introduction, one of our example applications uses the notion of generalized similarity to find members of *ad-hoc categories*. Given several phrases, it averages together their corresponding vectors in ConceptNet and returns other concepts that are sufficiently similar. This technique can fill in the missing members of a category, much like the proprietary Google Sets (`http://labs1.google.com/sets`).

The combination of ConceptNet and AnalogySpace provides a broad base of domain-general knowledge that can benefit AI systems, particularly those that work with natural language. A large corpus alone, or an inference technique alone, would not accomplish this, but in combination they give access to a wealth of knowledge.

Acknowledgments

We would like to thank James Pustejovsky for all of his help. Additional thanks go to Jason Alonso and Kenneth Arnold for their work on Divisi, the software package which can be downloaded for use in working with AnalogySpace. The Common Sense Computing Initiative would like to thank the tens of thousands of users whose contributions have made the Open Mind project possible, as well as the many people who volunteered to take our evaluation.

References

Banerjee, S., and Pedersen, T. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *"Eighteenth International Joint Conference on Artificial Intelligence"*.

Chklovski, T. 2003. Learner: a system for acquiring commonsense knowledge by analogy. In *K-CAP '03: Proceedings of the 2nd International Conference on Knowledge Capture*, 4–12. New York, NY, USA: ACM Press.

Coursey, K. 2007. WAC: Weka and Cyc: Teaching Cyc to learn through self-recursive data mining. *In Proceedings of the Workshop on Common Sense and Intelligent User Interfaces*.

Deerwester, S. C.; Dumais, S. T.; Landauer, T. K.; Furnas, G. W.; and Harshman, R. A. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41(6):391–407.

Eslick, I. 2006. *Searching for Commonsense*. Ph.D. Dissertation, MIT Media Lab.

Gupta, R., and Kochenderfer, M. J. 2004. Using statistical techniques and WordNet to reason with noisy data. In *Workshop on Adaptive Text Extraction and Mining, Nineteenth National Conference on Artificial Intelligence (AAAI-04)*.

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

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

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

Lieberman, H.; Faaborg, A.; Daher, W.; and Espinosa, J. 2005. How to wreck a nice beach you sing calm incense. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*.

Liu, H., and Singh, P. 2004. ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal* 22(4):211–226.

Patwardhan, S., and Pedersen, T. 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *EACL 2006 Workshop Making Sense of Sense—Bringing Computational Linguistics and Psycholinguistics Together*, 1–8.

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

Schubert, L. K. 2002. Can we derive general world knowledge from texts? *In Proceedings of the Human Language Technology Conference*.

Singh, P.; Lin, T.; Mueller, E. T.; Lim, G.; Perkins, T.; and Zhu, W. L. 2002. Open Mind Common Sense: Knowledge acquisition from the genera l public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/Coop IS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 200 2*, 1223–1237. London, UK: Springer-Verlag.

Speer, R. 2007. Open Mind Commons: An inquisitive approach to learning common sense. *Proceedings of the Workshop on Common Sense and Interactive Applications*.

Suh, S.; Halpin, H.; and Klein, E. 2006. Extracting common sense knowledge from Wikipedia. In *Proc. of the ISWC2006 Workshop on Web Content Mining with Human Language technology*.

Turney, P. D. 2005. Measuring semantic similarity by latent relational analysis.

Wall, M. E.; Rechtsteiner, A.; and Rocha, L. M. 2003. *A Practical Approach to Microarray Data Analysis*. Norwell, MA: Kluwel. chapter 5, 91–109.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition.