Using Analogy To Acquire Commonsense Knowledge from Human Contributors

by

Timothy A. Chklovski

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2003

Author .....
Department of Electrical Engineering and Computer Science
January 31, 2003

Certified by....
Patrick H. Winston
Ford Professor of Artificial Intelligence and Computer Science
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Using Analogy To Acquire Commonsense Knowledge from Human Contributors

by

Timothy A. Chklovski

Submitted to the Department of Electrical Engineering and Computer Science
on January 28, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The goal of the work reported here is to capture the commonsense knowledge of non-expert human contributors. Achieving this goal will enable more intelligent human-computer interfaces and pave the way for computers to reason about our world. In the domain of natural language processing, it will provide the world knowledge much needed for semantic processing of natural language.

To acquire knowledge from contributors not trained in knowledge engineering, I take the following four steps: (i) develop a knowledge representation (KR) model for simple assertions in natural language, (ii) introduce *cumulative analogy*, a class of nearest-neighbor based analogical reasoning algorithms over this representation, (iii) argue that cumulative analogy is well suited for knowledge acquisition (KA) based on a theoretical analysis of effectiveness of KA with this approach, and (iv) test the KR model and the effectiveness of the cumulative analogy algorithms empirically.

To investigate effectiveness of cumulative analogy for KA empirically, LEARNER, an open source system for KA by cumulative analogy has been implemented, deployed,[1] and evaluated. LEARNER acquires assertion-level knowledge by constructing shallow semantic analogies between a KA topic and its nearest neighbors and posing these analogies as natural language questions to human contributors.

Suppose, for example, that based on the knowledge about "newspapers" already present in the knowledge base, LEARNER judges "newspaper" to be similar to "book" and "magazine." Further suppose that assertions "books contain information" and "magazines contain information" are also already in the knowledge base. Then LEARNER will use cumulative analogy from the similar topics to ask humans whether "*newspapers* contain information."

Because similarity between topics is computed based on what is already known about them, LEARNER exhibits bootstrapping behavior — the quality of its questions improves as it gathers more knowledge. By summing evidence for and against posing

[1]The site, "1001 Questions," is publicly available at http://teach-computers.org/learner.html at the time of writing.

any given question, LEARNER also exhibits noise tolerance, limiting the effect of incorrect similarities.

The KA power of shallow semantic analogy from nearest neighbors is one of the main findings of this thesis. I perform an analysis of commonsense knowledge collected by another research effort that did not rely on analogical reasoning and demonstrate that indeed there is sufficient amount of correlation in the knowledge base to motivate using cumulative analogy from nearest neighbors as a KA method.

Empirically, evaluating the percentages of questions answered affirmatively, negatively and judged to be nonsensical in the cumulative analogy case compares favorably with the baseline, no-similarity case that relies on random objects rather than nearest neighbors. Of the questions generated by cumulative analogy, contributors answered 45% affirmatively, 28% negatively and marked 13% as nonsensical; in the control, no-similarity case 8% of questions were answered affirmatively, 60% negatively and 26% were marked as nonsensical.

Thesis Supervisor: Patrick H. Winston
Title: Ford Professor of Artificial Intelligence and Computer Science

Acknowledgments

Many brilliant and hard working people have helped make this a better work. I thank the committee — Patrick Winston, Randall Davis, Peter Szolovits and David Stork, for teaching me to be a serious scientist.

Patrick Winston deserves particular thanks for making me always keep in mind the high-level vision of this work, Randall Davis for teaching me to be thorough and for being particularly thorough in the constructive criticism of the work in progress, and Peter Szolovits for taking extra time in the discussions that shaped the thoughts and themes underlying the thesis.

I also wish to thank David Stork for his leadership in launching the broad Open Mind initiative, for his attention to this thesis beyond the call of duty and for his numerous helpful, thorough and insightful comments. I thank Ricoh Innovations for providing "Open Mind" T-shirts for rewarding exceptional contributors to LEARNER.

I thank Marvin Minsky, advisor of my Master's research, for encouraging me to ignore what everyone else thinks and instead work on the problem that I believe to be the most important.

I thank my wife, Tara Chklovski, for her boundless support, limitless optimism, tireless willingness to hear about the thesis, and selfless proofreading. She has made the tough days easier and the good days amazing.

I also thank my parents, Anatoli and Lucy Chklovski, whose limitless care and support have made this work possible. It is in conversations with Anatoli that the approach of cumulative analogy was originally outlined and its merits were formulated. I thank him greatly for all the invaluable in-depth discussions about the thesis, and the discussions on all the related and unrelated topics.

I thank Push Singh for many discussions of methods of public acquisition of knowledge, for sharing his experiences in implementing Open Mind Commonsense, and for making the data gathered with that effort publicly available (a subset of that knowledge base formed the seed knowledge base in the LEARNER).

I deeply thank Matthew Fredette for contributing his brilliant programming. He

has contributed much of the supporting and infrastructure code for LEARNER on top of FramerD and has also contributed to the integration of external packages used, FramerD and Link Grammar Parser. I thank Kenneth Haase, whose FramerD database LEARNER uses, for his development and support of FramerD. I also thank Davy Temperley, Daniel Sleator, John Lafferty for implementing Link Grammar Parser and making it available for research purposes.

I thank Erik Mueller and Kathy Panton for making available some data on existing knowledge bases. Some data provided by them are reproduced in this thesis.

I thank Rada Mihalcea for her comments on the thesis, advice on approaching the issue of lexical ambiguity, and for providing some data on evaluation of Open Mind Word Expert. My discussion of approaches to WSD in Section 6.4.2 is inspired by Rada's treatment of the topic. I thank Deb Roy for discussions of the relationship between natural language assertions and other possible representations of meanings of concepts.

All of these people have helped make this thesis a better piece of work. Any inaccuracies or errors or shortcomings of this work are my responsibility.

I would like to thank the National Science Foundation for the three years of fellowship support they provided, and the MIT Electrical Engineering and Computer Science department for supporting me for one term with the Edwin S. Webster Graduate Fellowship in Electrical Engineering and Computer Science.

I also thank Marilyn Pierce and Peggy Carney, administrators of the EECS Graduate Students Department, for their attention and help far beyond the call of duty with the administrative aspects of the degree, as well as Anthony Zolnik, Fern DeOliveira and Nira Manoharan for their administrative help in scheduling meetings with the thesis committee, reserving facilities, and so on.

Finally, I thank the contributors that have contributed their knowledge to the 1001 Questions web site, turning LEARNER from an experimental system into a source of commonsense knowledge. Particular thanks go out to the contributors who have left comments, suggestions, and other feedback about the project as a whole.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Knowledge acquisition bottleneck

For decades now, the field of Artificial Intelligence (AI) has recognized the need for representing knowledge that people have in a form that computers can use. *Knowledge acquisition* (KA) is a task in AI concerned with eliciting and representing knowledge of human experts so that it can later be used in an application, typically a knowledge-based system.

One of the most fundamental and still unsolved problems in knowledge acquisition goes by the name of *knowledge acquisition bottleneck*. The term, coined by Feigenbaum (Feigenbaum, 1984), refers to the difficulty of collecting from contributors data sets containing knowledge of sufficient level of detail and subtlety. Since that time, the advent of the web has made billions of documents available for near-instantaneous access, helping people share information with each other. This glut of human-readable data, however, has not yet translated into a windfall of machine-understandable data. Rather, it has only exacerbated the need for *machine-understandable* data to help process all the ordinary documents.

In addition, I believe the acquisition bottleneck also stems from the desire to capture perfectly unambiguous and non-contradictory knowledge at acquisition time. While such knowledge seems easier to work with, it is very difficult to formulate and add. Lack of methods for handling ambiguous and contradictory knowledge is, in my

view, the other part of the problem.

This thesis attacks the acquisition bottleneck for commonsense knowledge by providing a framework for collecting large datasets from many independent non-expert human contributors. To this end, I formulate *cumulative analogy*, a class of algorithms that formulate plausible new assertions about objects in the knowledge base. I test cumulative analogy empirically by implementing, deploying and evaluating LEARNER, a system for formulating and selecting the appropriate questions to pose to contributors who are not expert knowledge engineers. This approach can enable acquisition of commonsense knowledge assertions from very large numbers (potentially millions) of contributors.

This thesis demonstrates the following:

> *Commonsense knowledge can be acquired from contributors not trained in knowledge engineering. High-quality knowledge acquisition questions can be formulated automatically by surface-level analogy that poses new questions based on the information already in the system; the employed analogical reasoning exhibits bootstrapping and noise canceling features.*

To make the notion of commonsense knowledge in the above statement more concrete, I cite several examples of the kind of assertions collected by LEARNER. Some assertions formulated by LEARNER using cumulative analogy and confirmed to be *true* by human contributors are as follows:

- Cars are useful tools
- A jar can hold a liquid
- A person can own a laptop
- Tomatoes are sold at stores
- Humans hold guitars when playing them

Some assertions that LEARNER has formulated as plausible hypotheses, but with the help of human contributors, has discovered to be *false* are as follows:

- A crab is a kind of fish
- An apple is a high fat food
- Salt becomes ice when frozen
- A germ is an animal
- A guitar is a brass instrument
- Dogs purr when stroked.

In addition to demonstrating applicability of analogical reasoning to knowledge acquisition, this work also aims to provide a growing collection of commonsense knowledge to the research community, as well as provide the platform for experimenting with analogical reasoning over the collected knowledge for purposes other than knowledge acquisition, for example, for answering questions about what is asserted and implied by the assertions gathered in the knowledge base.

1.2 Motivation

In this section, which consists of four parts, I motivate both the goal and the methodology of the work presented here. In Section 1.2.1, I motivate the need for creating large knowledge repositories of commonsense knowledge. In Section 1.2.2, I describe some important real-world goals that a system such as LEARNER can help accomplish. In Section 1.2.3, I motivate the approach of acquiring commonsense knowledge from human contributors (as contrasted with, for example, attempting to automatically extract such knowledge from textual corpora). Finally, in Section 1.2.4, I argue that research in sophisticated knowledge acquisition techniques, being both incremental and "AI-complete" in the limit, is a good way to make progress towards "hard AI" (human-level knowledge and reasoning ability in a constructed system).

1.2.1 The case for creating a commonsense knowledge base

Knowledge acquisition techniques enable construction of knowledge bases. Yet, is it important to have a knowledge base of commonsense knowledge at all? I believe

creation of a large, publicly available commonsense knowledge base (together with methods for reasoning over such a knowledge base) is very important for attaining a number of both research and practical goals. In this section, I cite the literature motivating creation of such knowledge bases and put forth additional arguments for importance of attaining this goal.

There is practical evidence of usefulness of WordNet, a lexical knowledge base that captures some commonsense knowledge (Fellbaum, 1998; Lenat, Miller and Yokoi, 1995). WordNet provides information about senses of words, as well as "is-a," "part-of," "antonym," and a few other relationships between senses of words. As such, it can be viewed as capturing a subset of commonsense knowledge (containing knowledge such as "an automobile is a kind of vehicle" and "a wheel is a part of a bicycle." Applicability of WordNet to human language technology and knowledge processing community "has been cited by more than 200 papers and implemented systems that have been implemented using WordNet" (Harabagiu, Miller and Moldovan, 1999).

At the same time, WordNet is far from an exhaustive source of codified commonsense knowledge (Lenat, Miller and Yokoi, 1995). It does not provide more extensive commonsense knowledge such as that represented by the CYC project (Lenat, 1995; Lenat and Guha, 1990). Lenat has motivated the need for the more extensive commonsense knowledge by pointing to need for it in a number of Natural Language Processing (NLP) tasks. For example, Lenat argues that commonsense knowledge is needed to address NLP tasks such as resolving pronoun reference and machine translation. As example of commonsense knowledge base helping resolve pronoun reference, consider resolving what "they" refers to in "The police arrested the demonstrators because they feared violence" and "The police arrested the demonstrators because they advocated violence" (Lenat, 1995). An example of importance of commonsense knowledge in machine translation Lenat cites is as follows. Consider translating the sentence "Mary poured the water into the teakettle; when it whistled, she poured the water into a teacup." Since Japanese does not provide a single word for liquid water, translating the above sentence requires substitution of the Japanese word for "cold water" for the first instance and the Japanese word for "hot water" for the second

(Lenat, 1995). Another argument by Lenat in favor of commonsense knowledge has to do with semantic interpretation of compound nominals. Consider the phrase "tree doctor." In the absence of commonsense knowledge, it is difficult, if not impossible, to interpret it as a person who treats trees rather than a tree that practices medicine (Lenat, Miller and Yokoi, 1995).

Additional arguments in favor of constructing large knowledge bases emerge from the community that tries to mine the knowledge from texts. For example, Moldovan and Gîrju state: "[The field's] inability to build large knowledge bases without much effort has impeded many AI developments" (Moldovan and Gîrju, 2001). As specific examples, they point to reliance of the most successful current information extraction (IE) systems on hand-coded linguistic rules, making these systems difficult to port to other domains. Moldovan and Gîrju also point to the leveling off of results obtained at the Message Understanding Conference (MUC), and cite the common sentiment in that field that further progress will not be possible without knowledge intensive tools that support commonsense reasoning (Moldovan and Gîrju, 2001). Finally, Moldovan and Gîrju point out the need for commonsense knowledge and the need for it to enable further progress in in question answering (Moldovan and Gîrju, 2001).

Perhaps the strongest argument in favor of creating a publicly available knowledge base of commonsense knowledge is the evidence that much research on natural language understanding is currently trying to work around the problem. In other words, the lack of commonsense knowledge is having a widespread effect on methods being developed, approaches being taken, and progress being made by researchers working in various areas of Natural Language Understanding (NLU).

"Commonsense knowledge" in the NLU and NLP communities is often referred to as "world knowledge." Lack of codified world knowledge shaping is having a widespread effect on the current research. I present two specific examples: one from the field of machine translation, and one from question answering. In machine translation, Dorna and Emele argue that a practical system should avoid disambiguation whenever possible because disambiguation needs world knowledge plus some ability to reason over it (Dorna and Emele, 1996).

In question answering, evaluation criteria and goals of research are being set lower, with unavailability of world knowledge (and mechanisms to carry out commonsense reasoning over it) being cited as the reason for backing off from the more ambitious task of reading comprehension (Schwitter, Moll'a, Fournier and Hess, 2000).

As further evidence of need for broad commonsense knowledge, the importance of developing large knowledge bases capable of providing shallow but broad knowledge about motives, goals, people, countries and conflict situations (a subset of commonsense knowledge) in conjunction with deeper specific domain knowledge has been recognized by DARPA's $34 million High Performance Knowledge Bases (HPKB) initiative that ran from 1996 to 1999 (Cohen, Schrag, Jones, Pease, Lin, Starr, Gunning and Burke, 1998), followed by the Rapid Knowledge Formation initiative of similar magnitude (DARPA, 2000).

As an additional argument for need for a commonsense knowledge base and a system such as LEARNER capable of reasoning over it and enlarging it, the next section lists some specific practical goals accomplishing which requires, among other technologies, a large commonsense knowledge base and a mechanism of reasoning over it.

1.2.2 Practical goals Learner helps achieve

The goals of the LEARNER system described here are similar to those of the CYC project, despite some important differences in methodology. In addition to the reasons described in the previous section, I believe implementing LEARNER is important because it will help to take us toward the following practical goals:

Self-service Help Desks. Gathering knowledge directly from contributors and integrating it with natural language processing can enable responding to customer support queries posed in natural language. Such knowledge repositories, coupled with mechanisms for inference over them can enable deployment of help desks that answer questions without duplication of human effort and with ever increasing precision, automating what is today still costly and cumbersome.

Such technology can also be applied in educational settings, taking FAQs and independent learning to the next level.

Voice Command and Control. The knowledge LEARNER gathers can enable controlling computers and other devices with your voice. Continuous speech recognition has made great advances in recent years (Lamel, Gauvain and Adda, 2001). The new challenge is: given the recognized speech, figure out what the user actually wanted to do, what commands should actually be executed.

When speaking to each other, people communicate effectively because the listener is assumed to have a lot of commonsense knowledge and some reasoning ability. Computers and other devices can potentially join the class of intelligent listeners if they are equipped with a large common sense knowledge base and can operate on that knowledge.

Databases for Heterogeneous Knowledge. Currently, databases serve the task of storing and manipulating *structured* knowledge. A lot of valuable knowledge is simply too heterogeneous to be stored in a structured format. That is in part why the looser organizational approach of the World Wide Web has been so successful. LEARNER can capture assertion-level information and be as useful on that level as the Web is on the document-level.

Under this approach, knowledge repositories can be created and grown without the need for designing a data schema. Such knowledge repositories would be able to capture exceptions and nuances of data in a very natural way. Rather than performing joins, these repositories would run inference operations over their data to respond to queries. Finally, they would work with natural language, eliminating the need for specialized programming to encode queries that extract data from a repository.

1.2.3 A comparison of approaches to collecting commonsense knowledge

In this section, I compare some plausible approaches to creating large commonsense knowledge bases and motivate the methodology of this thesis, namely approaching the problem by collecting knowledge from contributors who are not knowledge engineers. I briefly compare the approach with having a team of knowledge engineers hand-craft a large knowledge base, and compare it more extensively with attempting to mine a knowledge base automatically from electronically accessible texts.

Creating a large knowledge base by having a team of knowledge engineers hand-craft it is perhaps the most straightforward approach. The prime example of this approach is the CYC project. The difficulties include the large cost and complexity of such an effort. The CYC project has consumed over two hundred man-years, but the original goal are yet to be reached (and have been restated as possibly overly optimistic) (Guha and Lenat, 1990; Lenat, 1995). Another datapoint on the amount of effort required in constructing a knowledge base is the knowledge base for Botany containing 20,000 concepts and 100,000 facts took Porter and staff ten years of development (Porter and Souther, 1999). At the same time, the approach of bringing knowledge engineers to bear on a problem has some undeniable advantages and should not be discounted. Such an approach provides the highest quality of knowledge, trading off the amount (or rate) of acquisition for quality.

In light of this "knowledge acquisition bottleneck," there is a significant desire in the knowledge acquisition community to address the bottleneck by acquiring knowledge via text mining. (As a discipline, text mining employs a combination of statistical and linguistic methods to extract information (often automatically) from large corpora of human-generated text. For the purpose of this discussion, I use the term interchangeably with "knowledge acquisition from texts.")

As discussed below, text mining may yield much useful knowledge. However, at least some researchers that work on issues of commonsense knowledge feel that much of the commonsense knowledge needed to understand texts is not actually written

down. Rather, the texts, even didactic texts such as encyclopedia, presume that the reader already has the commonsense knowledge needed to understand them (Lenat, Guha, Pittman, Pratt and Shepherd, 1990),(Nilsson, 1995, p. 14).[1] Similar sentiment has been expressed in the context of a discussion of the commonsense knowledge necessary to construct intelligent computer interfaces:

> Much of our commonsense knowledge information has never been recorded at all because it has always seemed so obvious we never thought of describing it. (Minsky, 2000).

A more focused discussion of what information is and is not present in dictionaries can be found in (Atkins, Kegl and Levin, 1986), who generally find that dictionaries omit much of the commonsense information humans know about objects. Dolan et al. point out that in addition to the main entry for a concept (e.g., "flower"), more information about a concept is also sometimes found in definitions of related concepts ("petals," "leafy plants," and so on). They also point out, however, that the their example is drawn from a dictionary for someone learning the language — of the approximately 40,000 concepts defined in it only about 2,500 appear in the bodies of definitions. This causes the dictionary to contain much information about the "core" 2,500 concepts, but little information about other concepts (Dolan, Vanderwende and Richardson, 1993).

Keeping in mind the caveat of a large portion of commonsense knowledge not being written down, I now overview some systems that aim to extract concepts as well as taxonomic and other relations between concepts. One of the earlier examples of such systems is Hearst's system for acquisition of ontological information from text (Hearst, 1992). The system extracts pairs of words in the hypernym, or "is-a" relationship by leveraging specific syntactic patterns in unrestricted text, such as "*noun1* such as *noun2*" where neither *noun1* nor *noun2* are unmodified by additional

[1] I believe there are two exceptions to the observation. Some simple commonsense knowledge is present in books for young children (which, however, usually rely extensively on illustrations to convey much of such knowledge). The second source, in small amounts, is texts discussing commonsense knowledge in which such knowledge is presented by way of example.

nouns. For example, when the system encounters the phrase "animals such as cats," it will extract the relationship "a cat is a kind of animal."

MindNet is another system which extracts semantic relationships. MindNet extracts approximately 25 semantic relationships including *Location, Part_of, Purpose, Hypernym, Time, (Typical_)subject, (Typical_)object,* and *Instrument* (Dolan, Vanderwende and Richardson, 1993; Richardson, Vanderwende and Dolan, 1993). To my knowledge, this represents the largest number of relationships extracted by a system that mines a closed class of relationships. Akin to Hearst's system, semantic relationships are extracted via specifically constructed heuristic rules that recognize "occurrence of syntactic and lexical patterns which are consistently associated with some specific semantic relation, such as instrument or location" (Dolan, Vanderwende and Richardson, 1993). The heuristic rules are designed for and the extraction is applied to a particular dictionary available in electronic format, Longman's Dictionary of Contemporary English (LDOCE).

KAT (Knowledge Acquisition from Text) is another system that acquires both concepts and some semantic relationships from text (Moldovan and Gîrju, 2001). Starting with seed concepts such as "stock market" and "employment," KAT applies lexical heuristics (created semi-automatically with some human intervention), syntactic patterns and filtering by humans to identify additional related concepts and semantic relationships between them such as *hypernymy* (Is-a), *influence*, *cause*, and *Equivalent.* Extraction is done from textual corpora such as newspapers. Extraction of both the concepts and the relationships between them is performed with humans in the loop to filter out nonsensical concepts and misidentified relationships.

Compared to knowledge bases crafted by experts, text mining systems represent another point in the quality vs. quantity trade-off, trading off precision of the collected data for volume. In my view, the empirically observed (to date) weakness in distinguishing the extracted knowledge from noise is the Achilles' heel of current systems that aim to acquire knowledge from text. Let us review some results from the three systems mentioned above.

Hearst (Hearst, 1992) does not report overall statistics on the quality of the extrac-

tion. Rather, she reports some pairs that the algorithm has extracted that exemplify some classes of noise that can trip up extraction systems. For example, Hearst's system extracted pairs such as "king is-a institution". Hearst attributes it to metonymy in texts (substitution of the name of an attribute or feature for the name of the thing itself, an example of metonymy being "the White House signed a bill"). Other pairs exemplify a more common problem of underspecification in text. Examples included taxonomic relations between "plot" and "device," "metaphor" and "device" and "character" and "device" (omitting that the device in question is a literary device). Another class of problematic relations extracted involved relations that may be context or point-of-view dependent, such as "Washington is-a nationalist" and "aircraft is-a target."

In MindNet, the overall precision of extracting its semantic relations from LDOCE is estimated by the authors to be 78% (with the margin of error of ±5%). About half of the extracted relations are of the type *Hypernym* (Is-a), and these were accurate 87% of the time. The *Part_of* relation was accurate only 15% of the time, and the remaining relations were accurate 78% of the time.

KAT explicitly relied on humans in the loop to help identify relevant concepts. Some were extracted fully automatically, while the remaining ones were passed to human judges for inspection. KAT's implementors report that the human intervention in accepting or declining the concepts took about 20 minutes to process roughly 1500 concepts. Of the 196 concepts (such as "financial market") that were discovered in text and were not found in any of the preexisting dictionaries that the system consulted, the automated procedure has identified 77 (39.2%) with precision of 90%. Human filtering of the concept candidates that the automatic procedure neither accepted nor discarded identified the remaining 119 concepts (60.7%). In contrast to the concepts, all of the relations extracted by KAT were subjected to human filtering. Of 166 candidate pairs, 64 were accepted and 102 were rejected by inspection, a process that took approximately 7 minutes (Moldovan and Gîrju, 2001).

Moldovan and Gîrju argue for the promise of automatic extraction. They feel that among others, addition of ability to handle "complex linguistic phenomena such as

coreference resolution, word sense disambiguation and others" as well as "incorporation of an elaborate process for pattern classification" will bring a system such as KAT closer to fully automatic acquisition. Their results, however, can be interpreted differently. Rather than viewing human intervention as a limiting factor of knowledge acquisition, it can be viewed as a powerful alternative solution to the unsolved problem of low-noise automatic acquisition of knowledge from texts.

To sum up, I believe the observations that much of commonsense knowledge is not written down, together with the noisiness of the knowledge that can be extracted with today's technology, militate in favor of use of human contributors. The case for collection of commonsense knowledge from human contributors is further bolstered by the option to distribute the collection to the potentially enormous population of "netizens" — volunteers contributing over the world wide web (Hearst, Hunson and Stork, 1999). Such an approach has humans as the direct source of knowledge; such an approach permits clarification and verification of knowledge by asking several contributors. At the same time, the potential number of contributors is enormous. As such, this approach represents a combination of quality and quantity that lies somewhere in between the other two methods discussed in this section — creation of knowledge bases by professional knowledge engineers and acquiring knowledge from text. For the reasons stated above, I believe it is a powerful approach on its own and it may also be useful in combination with the other two. As Edward Fredkin, former director of MIT Laboratory of Computer Science once said: "the best way to find out the answer to a question is to ask someone who knows the answer."

1.2.4 Research in knowledge-based KA contributes to broader progress in AI

In this section, I argue that research in formulating knowledge acquisition questions by reasoning over already collected knowledge and posing these questions to a population of human contributors provides a viable path to getting at human-like reasoning, the core of artificial intelligence. Here are the reasons supporting this stance:

You can start simple

A KA system can be made operational with very simple acquisition mechanisms in place. At a later point, more sophisticated ones mechanisms can be swapped in. Extensions of a simple uniform reasoning mechanism can include adding internal representations for notions of space and time, adding ability to handle simple arithmetic reasoning, and other competences or "agencies", in the sense introduced by Minsky (Minsky, 1986).

You get feedback quickly

The performance is "directly evaluable" — the amount of knowledge being collected and the quality of the questions posed using a specific reasoning mechanism can be tracked easily. For example, if a certain mechanism is overly ambitious in generalizing, it can be observed through it making a lot of incorrect (and hence corrected by contributors) predictions.

You need to solve the hard problems to do it well

While having a low entry barrier, the task is difficult to do perfectly. Arguably, it is "AI-complete" (which, analogously to "NP-complete," means that a full solution is equivalent to solving other AI-complete problems such as Machine Vision or Natural Language Understanding). Posing knowledge acquisition questions well requires analysis of prior knowledge to determine what new questions to ask, and evaluation the incoming in light of the prior knowledge. In a way, figuring out how to *gather* knowledge is also addressing how to effectively *use* the knowledge.

Many can contribute

Deploying a knowledge acquisition system online makes the challenges of creating an understanding system very visible to a broader public. I believe that making such a system publicly available can help provide a common testbed for various approaches in AI. Even more importantly, has a shot at attracting more people not in the mainstream of the field to contributing knowledge to the KA system, to extending the system's mechanisms, and ultimately to working on

artificial intelligence. In a sense, a parallel can be drawn between the resulting process of bootstrapping an increasingly intelligent KA system and the bootstrapping of its scientific knowledge performed by a modern human society as a whole.

1.3 Structure of this document

Chapter 2 presents a high level example of how LEARNER — the implemented cumulative analogy KA system — operates, Chapters 3 (Representation), 4 (Algorithms), and 5 (Interface) explain how LEARNER works. They present, respectively, (i) the overall architecture of LEARNER and the knowledge representation scheme used, (ii) the algorithms that operate on this representation to pose knowledge acquisition questions, and (iii) the interface that presents the questions and provides users with feedback about the impact made by the knowledge they have just added. Chapter 6 discusses the kinds of ambiguity in the collected knowledge, the sets of tasks for which ambiguity is and is not problematic, and suggests ways to reduce such ambiguity.

Chapter 7 (The Correlated Universe) provides a theoretical analysis of effectiveness of cumulative analogy, motivating its use for knowledge acquisition by studying the amount of correlation between assertions in a subset of a commonsense knowledge base collected by Singh (Singh, Lin, Mueller, Lim, Perkins and Zhu, 2002).

Chapter 8 presents experimental evaluation of effectiveness of knowledge acquisition by cumulative analogy and reports on the kinds of knowledge collected. Chapter 9 consists of three sections. The first section overviews some of the relevant prior work from the expert system, machine learning, knowledge acquisition interface, and text mining traditions. The next section discusses how LEARNER can be improved further, including improvements to the process of measuring similarity as well as sketching a generalization of cumulative analogy. Ways to improve the resulting knowledge base are also briefly discussed. Finally, and perhaps most importantly, Section 9.3 distills the contributions of the thesis, summarizing what this work has demonstrated.

Chapter 2

Examples of Acquiring Knowledge in Learner

I describe the details of the implementation of the system in Chapters 3 through 5. In this chapter, I introduce the sort of analogy that LEARNER performs, present examples of how LEARNER operates, and provide a screenshot of the acquisition interface.

A clarification of my use of the term "analogy" is in order. When writing "make an analogy" or "by analogy," I do not refer to analogies of the sort found on standardized tests, such as

`hat:head::glove:hand`,

or "hat is to head as glove is to hand." Neither do I refer to the more elaborate kind of analogy that arises from aligning graph-like structures (Gentner, 1987; Winston, 1980). Rather, I refer to "analogy" as a process of inference that maps assertions about objects onto another (typically similar) object. This usage is consistent with the first definition of analogy provided by the Merriam-Webster dictionary:

> **Analogy**: "Inference that if two or more things agree with one another in some respects they will probably agree in others."

This usage is also consistent with the usage in scientific literature. For example, a work on assessing the role of analogy in Natural Language Processing (Federici,

Montemagni and Pirrell, 1996) provides the following definition for the process of "generalization by analogy":

> "Generalization by analogy can be defined as the inferential process by which an unfamiliar object (the target object) is seen as an analogue of known objects of the same type (the base objects) so that whatever properties are known about the latter are assumed to be transferable to the former."

By "generalization by analogy," I shall mean, formally, if O and O' are objects and P_1 and P_2 are properties, then:

If O has property P_1, and
O' has property P_1, and
O has property P_2, then it can expected that
O' may have the property P_2.

Let us consider a simple example. Imagine that the system already knows the following assertions about (or, equivalently, properties of) computer mice and keyboards:

```
a computer mouse has buttons
a computer mouse fits in your hand
a computer mouse helps you use a computer
a computer mouse attaches to the computer with a cable

a keyboard helps you use a computer
a keyboard attaches to the computer with a cable
```

Based on these statements, the system would decide that keyboards are similar to mice (because they both "`help you use a computer`" and "`attach to the computer with a cable`") and, *by analogy*, map additional statements known about mice onto keyboards. As a result, it may come up with the following hypotheses:

`a keyboard has buttons`

`a keyboard fits in your hand`

It is then up to the human contributor to clarify that yes, "a keyboard has buttons" (or to enter a modified assertion "a keyboard has keys," but that "a keyboard does not fit in your hand." In this example, knowledge about "`computer mice`" (a *source topic*) allowed us to pose questions about "`keyboards`" (the *target topic*). As described in the following chapters, the implemented LEARNER algorithm uses multiple source topics for any given target topic, summing "pro" and "con" evidence from each source topic when evaluating whether to pose any given question about the target topic.

Given a target topic selected by the contributor, the system presents a number of multiple-choice questions. A sample screenshot is presented in Figure 2-1. The snapshot of the web page shows LEARNER posing questions about "`newspaper.`" Comparing what is known about "`newspaper`" with what is known about other topics has determined that newspaper (when the snapshot was taken), was most similar to "`book,`" "`map,`" "`magazine,`" and "`bag.`" These are shown in the line beginning with the words "Similar topics." From these, properties are mapped onto "`newspaper,`" and presented as questions in text boxes. For example, the first question shown is "newspapers contain information?"

Contributors can reply to these questions by selecting an answer such as "Yes" or "No" from the drop-down box next to the question. Contributors are also free to modify the question prior to answering it. For example, the contributor may choose to modify the above question to read "newspapers contain *recent* information?" and reply to this modified question. This ability to modify and enter new information keeps expanding the potential set of questions LEARNER can pose. See Section 5 for a more in-depth explanation of the details of the interface.

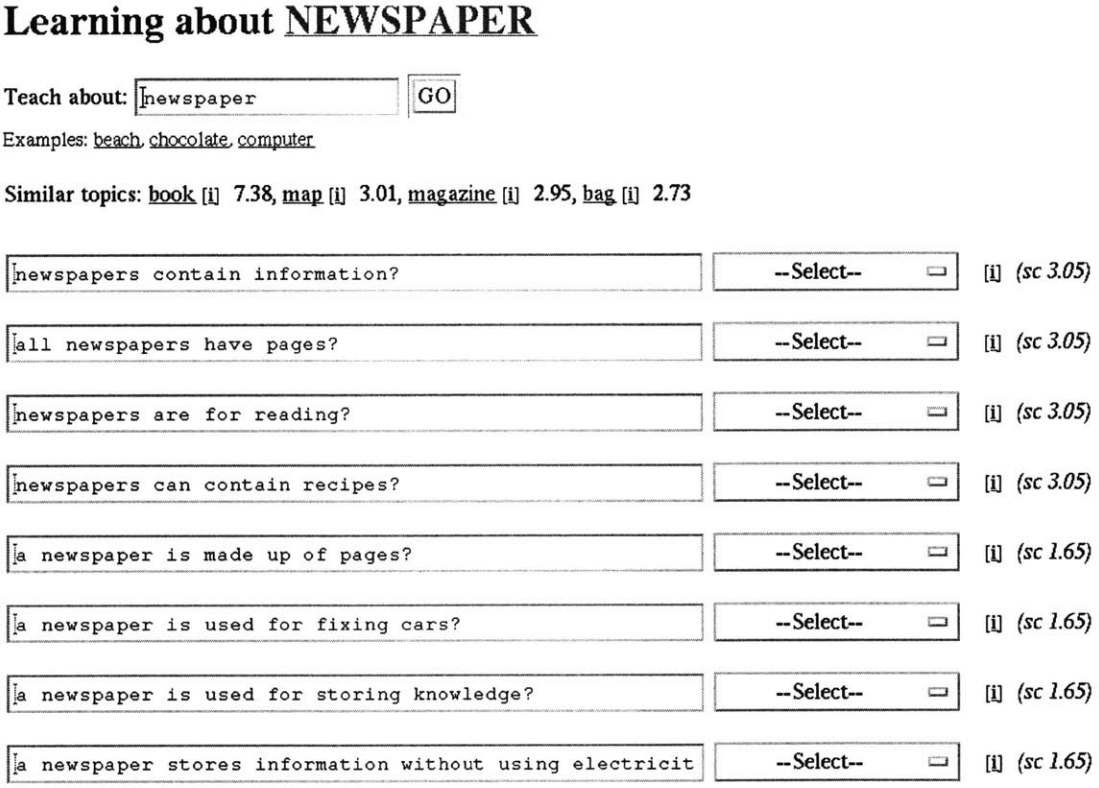


Figure 2-1: Screenshot of acquiring knowledge about "newspaper."

Chapter 3

Representation

This and the following two chapters explain how LEARNER works. They present the knowledge representation scheme used, the algorithms that operate on this representation to pose knowledge acquisition questions, and the interface that presents the questions and provides users with feedback about the impact made by the knowledge they have just added.

The seed knowledge base that LEARNER was launched with is presented in Section 7.1. Further analysis of kinds and amount of knowledge collected as a result of running the system is given in Section 8.3. The current chapter covers the form of knowledge LEARNER accepts (Section 3.2) and how this knowledge is represented internally (Sections 3.3 and 3.4).

3.1 Overall architecture

In this section, I introduce the overall architecture of the system. Figure 3-1 specifies the existing technologies used in LEARNER and their overall arrangement. LEARNER uses components developed by other researchers:

- Link Grammar Parser (Sleator and Temperley, 1993; Temperley, Sleator and Lafferty, 2000) as the underlying parsing technology.

- FramerD (Haase, 1996) for the storage model and the language in which LEAR-

NER is implemented.

- WordNet (Fellbaum, 1998), a lexical knowledge base, for tokenization of noun phrases and "is-a" information about nouns.

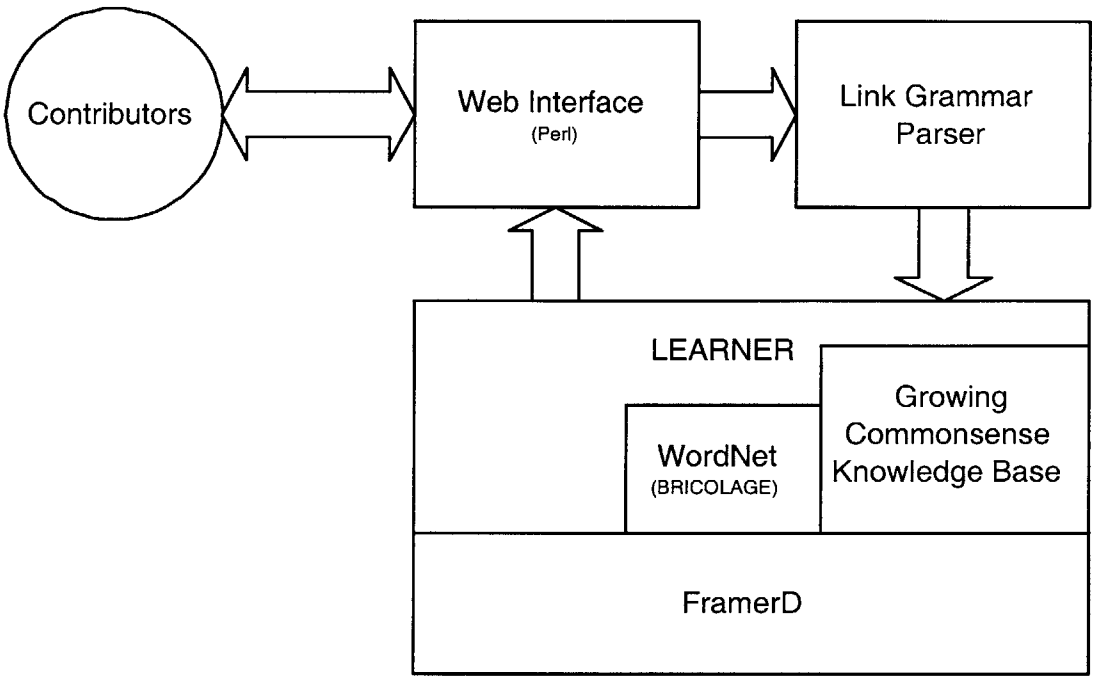


Figure 3-1: Overall architecture of the LEARNER knowledge acquisition system.

Details about Link Grammar Parser and FramerD can be found in Appendices A and B, respectively. Contributors interact with LEARNER using a web CGI interface implemented in Perl. Contributor input is processed with the Link Grammar Parser, and the parsed assertions are passed to the core LEARNER component. Excluding the Web interface, LEARNER has been implemented in FDscript, a freely available variant of Scheme with integrated access to object oriented database (FramerD) and additional features such as simulation of nondeterministic computation.

On top of FramerD, the two additional components that LEARNER draws upon are WordNet and the growing commonsense knowledge base. WordNet is a well-known lexical knowledge base; its use is detailed in Section 4.2.3. WordNet information has been stored as a FramerD database; WordNet is distributed in this format as part of the BRICOLAGE package available with FramerD. The collection of commonsense

knowledge that LEARNER uses in posing knowledge acquisition questions consists of a seed knowledge base extracted by me from the knowledge collected by a different knowledge acquisition effort (Singh, 2002; Singh, Lin, Mueller, Lim, Perkins and Zhu, 2002) (as described further in Section 7.1). The knowledge acquisition questions and other output generated by LEARNER (such as results of searching the knowledge base for assertions containing a specific term) are passed back to the Web interface which presents this information back to the contributors.

The code for the project is open source. At the time of writing, everything needed to install, run and modify a copy of the system is available at

http://sourceforge.net/projects/learner.

The reader is encouraged to experiment with and extend the system.

3.2 Form of accepted knowledge

LEARNER imposes no restrictions on the domain of textual knowledge it operates on. There are, however, a number of restrictions on the form of such knowledge. To allow volunteers to add knowledge without any special training in description logics or knowledge representation languages, LEARNER accepts input in the form of assertions in English.

LEARNER is designed to handle only single-sentence syntactically valid assertions, each of which is interpretable by a human reader on its own, in isolation. Furthermore, LEARNER is designed to handle primarily general factual knowledge about *generic concepts* (classes of objects), rather than specific events or individual objects. Here are some examples of assertions LEARNER *is* designed to handle:

- "swans are white,"
- "a hammer is a tool,"
- "a yacht has a sail,"
- "keys can unlock locks,"

- "computers are made up of parts."

The system expects each assertion to be one (possibly qualified) piece of information, not a conjunction of several. Disjunctive assertions are not supported, either. Here are some examples of assertions LEARNER is *not* designed to handle:

- "it is round" *(needs context)*,
- "John likes sandwiches" *(concerns an individual, not a class of objects)*,
- "a shopper bought a carrot" *(concerns a specific event)*,
- "a yacht has a sail and an anchor" *(a conjunction of two assertions)*,
- "John thinks that Mary knows that cars have wheels" *(second-order logic)*.

As the last example suggests, LEARNER is not designed to handle "assertions about assertions," — assertions that require second order logic when represented in a formal logic.

Because human-level understanding of the assertions has not been achieved, the above constraints cannot be enforced perfectly. For example, assertions concerning specific events such as "a professor cut grass on his lawn" are quite difficult to distinguish from general assertions such as "lawnmowers cut grass". Adding knowledge which lies outside of the system's intended scope is likely to result in generation of a larger number of strange or nonsensical knowledge acquisition questions.

There are, however, some enforced syntactic constraints that limit what assertions can be added. These constrains require that an assertion:

Be parsable — LEARNER uses the Link Grammar Parser internally; only sentences accepted by this parser are admitted. The Link Grammar Parser, developed by researchers at CMU (Sleator and Temperley, 1993; Temperley, Sleator and Lafferty, 2000), implements the "link" theory of grammar and is described briefly in Appendix A.

Be declarative — In particular, imperative statements (such as "go clean your room") are disallowed. This constraint is also implemented by analyzing the parser's output.

Not be in past, past perfect, or present perfect tenses — This serves to disallow assertions about specific events. The constraint is implemented by analyzing the parser's output.

Not start with referential words — Sentences starting with words such as "it," "this," "that," "these," "my," "I," and "both" are disallowed because such sentences, usually indicate that the sentence either requires a context (the referent for the anaphora), is about a specific object (as in "that man has a knife").[1]

Not contain conjunctions or disjunctions — Many conjunctions can instead be entered as two or more statements. This constraint is implemented by analyzing the parser's output.

As will become clear from the more technical description that follows, LEARNER addresses primarily collecting knowledge about objects and their properties. There are other segments of the full spectrum of commonsense knowledge that deserve further attention but which lie largely outside the scope of this work. The kinds of knowledge not covered or addressed only peripherally include:

- knowledge about "verbs" — knowledge about goals and effects of actions, sequence of subevents (scripts) for actions, knowledge about required tools as well as spatial and temporal preconditions for actions,
- knowledge about causes — "why can birds fly?",
- knowledge about typical motives and actions taken by people specific to more narrow contexts (e.g. "a person usually stands on a chair to change a light bulb"),
- knowledge about how knowledge is to be combined — "ostriches can run," but "dead things cannot run", so the answer to whether "dead ostriches can run" is unclear.

[1] In the reported experiments, sentences beginning with the word "the" have also been disallowed (to exclude such assertions as in "the building is tall"). However, based on contributor feedback and experience with the system, sentences beginning with "the" (such as "the heart pumps blood", "the Earth is round", and "the lion is one of the largest cats") have since been allowed again.

I believe that knowledge about objects and their properties, collection of which is tackled by the current system, is one of the more fundamental kinds of knowledge. Collecting this knowledge can lay the foundation on top of which collection of other kinds of knowledge (such as knowledge about causes) can be reasoned about and thus intelligently collected.

3.3 Sentences and their signatures

This and the following sections describe how natural language statements are processed into assertions to be stored in the knowledge base. Each sentence has a dual representation: (i) a *link* representation, which constitutes the parser's output from which the original string can be reconstructed. Note that the parser used in this work, Link Grammar parser (Sleator and Temperley, 1993), normally parses sentences into a set of links between words rather than producing a parse tree. Link output is more informative than a typical parse tree, providing more information about the relationships that hold between words in a sentence. and (ii) the *signature* representation produced from the link representation ("signature" is a term I introduce; it is not standard in the literature. The signature of an assertion is a canonical form that abstracts away syntactic details while capturing — well enough for the purposes of the system — what the assertion states.

The link representation is used to construct new assertions by substituting words in existing assertions. When a new knowledge acquisition question is generated, it is this representation that is used to ensure number agreement between subject and main verb, that the right indefinite article ("a" or "an") is used, and so on. See Appendix A for details about the parser, and Appendix C for details on the syntactic processing performed by LEARNER to generate knowledge acquisition questions.

All other processing of knowledge by the system, most importantly retrieval and matching, is done over signatures and data structures derived from them. In this section, I describe how a signature is derived from a sentence. In the next section, I detail how a sentence and its signature are transformed into an assertion about an

object.

The signature of a sentence is the set of nouns, verbs, adjectives (together with their parts of speech) that appear in subject, main verb, object(s) and prepositional phrases of a sentence, as well as adverbs in adverbial phrases. A signature is meant to preserve its most important information. In the signature, the base form of the word appearing in the sentence is used (the singular form for nouns, the infinitive for verbs). Determiners (such as "a," "the," "some" and so on) and closed-class words (such as "of," "with," "on," "if," and so on) are omitted. For efficiency of comparing signatures, signatures are stored as sorted lists of words (with their parts of speech), without preserving the order. (See the next section for an explanation how "an elephant pushes a cart" and "a cart pushes an elephant" result in different *assertions.*)

Because of this method of computing signatures, variations such as "a dog barks," "a dog can bark," "all dogs bark" will all have the same signature: "{dog_{noun}, bark_{verb}}".

3.4 Phrases, properties and assertions

Many declarative sentences can be viewed as an assertion in more than one way. Most declarative sentences accepted by LEARNER can be viewed at least as assertions about the sentence's syntactic *subject.* For example, "cats eat mice" asserts that "`cats`" "`eat mice.`" Some sentences, (such as "cats eat mice"), can also be viewed as assertions about their syntactic *object*: "`mice`" have the property that "`cats eat`" (them). Note that not all sentences have a noun phrase object (e.g. "cats are beautiful" does not). Finally, some sentences can also be viewed as assertions about their prepositional phrases. For example, "people eat with forks" can be viewed as asserting that a "`fork`" is something "`people eat with.`" These sentences may or may not have a direct syntactic object.

In all, three important kinds of phrases which may be present in a sentence can be identified: `subj` (subject), `obj` (object), and `pp` (prepositional) phrases. These,

when present, give rise to subj-, obj-, and pp-assertions. Each sentence is interpreted as an assertion about each kind of phrase present in it. For example, "cats eat mice" is interpreted as both as a (subject) assertion about "`cat`" and an (object) assertion about "`mouse.`" Every assertion has two parts: the object O (not to be confused with the sentence's syntactic object) about which something is asserted, and the property P being asserted about this object. I denote an assertion that an object O has the property P as $A(O, P)$. The phrase (one of subj, obj, or pp) that gave rise to this assertion is not included in the notation and is not important unless explicitly specified.

To create a subject assertion from a sentence, the sentence's signature and its syntactic subject (the nouns and adjectives making up the subject noun phrase) are extracted. The order of words in the subject is preserved, but the words are converted to canonical form (i.e. nouns are singularized). The singularized noun phrase becomes the subj-assertion's object O. The property P is computed by taking the sentence's signature, and removing from it all words present in the object. As in signatures, the order of words in properties is not preserved. Obj-assertions and pp-assertions are created similarly, using the corresponding phrase in forming the assertion's object.

When creating assertions from sentences, LEARNER attempts to handle negation correctly. Syntactic analysis of the sentence looks for "not" negating the meaning of the sentence. Here are some examples of negation that is correctly recognized:

```
cats do not fly
cats don't fly
a cat cannot fly
```

To represent the recognized negation, each assertion in the knowledge base has one of two *truth values*: $Tv = 1$ or 0. When referring to a specific object-property pair O_i and P_j, I will denote these as $Tv(O_i, P_j) = 1$ and $Tv(O_i, P_j) = 0$, and informally refer to assertions in these two classes as "is true" and "is false" assertions, respectively.

Because each assertion consists of an object and a property, the entire knowledge base can be visualized as a large matrix, with every known object of some asser-

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		x		x	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

Figure 3-2: Example of a matrix encoding assertions about objects and their properties. The actual matrix has tens of thousands of rows and columns. For simplicity, only "is true" subject-assertions are shown, marked by 'x'es. Blank cells denote that truth values for these assertions are not explicitly known.

tion being a row and every known property being a column. See Figure 3-2 for an illustration of an objects-properties matrix depicting a set of assertions.

Each sentence is assigned a unique ID that can be retrieved given the string of the sentence. The assertions that the sentence encodes are indexed by this ID, allowing for rapid retrieval of all assertions a sentence encodes, as well retrieval of some additional information such as phrases present, and the head noun of each phrase present.

Additionally, assertions are indexed by phrases and properties. For example, it is possible to quickly retrieve all subj-assertions with the object O being "`cat`," or all subj-assertions with the property "`have tail`."

Chapter 4

Algorithms

In this chapter, I present the principles that guided the formulation of the algorithms for question generation, the algorithms themselves, **Select-NN** (for "select nearest neighbors") and **Map-Props** (for "map properties"), and algorithms to further refine the set of generated questions.

4.1 Guiding principles

The goal of LEARNER is to acquire knowledge from human volunteers. What features should an approach to the problem of knowledge acquisition have? I discuss both methodological and architectural considerations.

Some top level methodological principles adopted in developing a solution are as follows:

- knowledge needs to be acquired *actively*. That is, the system should direct the acquisition to the knowledge not already present (DARPA, 1998).
- knowledge, especially in large multi-purpose knowledge bases, should be acquired incrementally (Menzies, 1998).
- knowledge needs to be maintained after it has been acquired. In other words, it is necessary to constantly analyze how acquired knowledge can be operationalized (used) and how it can be faulted (when does using it produce undesirable

or incorrect results), and what actions may be needed to correct the behavior (e.g., adding new knowledge to address the limitations of existing knowledge or correcting the already acquired knowledge) (Menzies, 1998).

- successful acquisition has to be *transparent.* That is, it should be possible to understand why the system did what it did, to see the impact your contribution made, and to correct behavior of the system. The importance of exposing the summary of the system's operations to a naive user is discussed by (Hellerstein, 1997). Hellerstein argues that in situations involving naive users and long processing times by the system, it is important to give the user access to a summary of the system's operation to reduce user frustration and provide a path for the user to learn how to best interact with the system.

The architectural desiderata are (i) modularity, so that many algorithms for question generation can be plugged in, and (ii) orthogonality, so that modules focus on their tasks without having to duplicate the same functionality.

To accommodate these goals, a kind of generate and test architecture has been employed. That is, the initial stage of processing generates questions, passing them to a set of modules that filter the questions, passing them to the output. I refer to the modules that propose questions as *generators* and to modules that combine and filter questions as *critics*, a term I use in the sense consistent with Minsky (Minsky, 1986) to refer to a more powerful version of a test. Unlike tests, critics are able to combine and modify their inputs in addition to merely filtering them.

Currently, LEARNER employs only one method for generating questions — by analogy. There are several filters being applied to questions generated. The filters remove questions that are non-grammatical, redundant, or those which are not likely to yield useful new knowledge.

Armed with these goals and architectural motivations, the following sections explain (i) the algorithms for generating questions by analogy, (ii) the filtering of the generated questions before they are presented to the user, and (iii) how the acquisition interface conforms to these goals.

4.2 Analogy

Generating knowledge acquisition questions by using analogy is at the heart of this thesis. This section presents the algorithms involved. A novel feature the analogical reasoning method employed in this work is that it relies on mapping properties from *many* similar objects, summing the evidence for posing any given question.

Because of its reliance on many sources of analogy, I call this method *cumulative analogy.* In this section, I present the algorithms for cumulative analogy and explains the algorithms' operation on a step-by-step example. (Recall that Ch. 2 presented a bird's-eye-view example of analogy in operation).

Given a knowledge acquisition topic O_{target} about which knowledge is being acquired, cumulative analogy is performed in two stages: (i) **Select-NN**, selecting the set $\mathcal{O}$ of nearest neighbors O_{src_i}, and (ii) **Map-Props**, projecting known properties of O_{src_i} back onto O_{target} and presenting them as questions.[1]

Both algorithms are explained on a simplified example of posing questions about "`newspaper`." The formal pseudocode for the two algorithms is also presented in Figures 4-5 and 4-8, respectively.

4.2.1 Select-NN

The details of **Select-NN** are as follows. First, Select-NN retrieves the properties of the target object, as illustrated in Figure 4-1.

Next, for each selected property, all assertions about this property and some other object with this property are identified, as shown in Figure 4-2. Both objects about which this property "is true" and "is false" are included.

Next, for every object that shares properties with "`newspaper`," the similarity between this object and "`newspaper`" are computed based on properties that they share and that they mismatch on (two objects mismatch on a property when a property is asserted as "is true" about one and as "is false" about the other). The contribution of

[1] It is the second step of projecting properties that motivates the terminology of "O_{src}" ("src" for "source") and "O_{target}."

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		**x**		**x**	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

Figure 4-1: Select-NN: Preparing to formulate questions about "`newspaper`." Properties already known about "`newspaper`" are identified.

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		**x**		**x**	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

Figure 4-2: Select-NN: All assertions about known properties of "`newspaper`" are identified, in this example the two being "`has pages`" and "`is for reading`."

each property to the total similarity score is weighted by this property's frequency in the entire knowledge base, with matches of rare properties receiving greater weight. The detailed formulas are given below.

As illustrated in Figure 4-3, up to ten most similar objects are selected. Prior to returning the objects, however, there is some additional filtering. No objects that are more general than the target object are returned. When there is more than one object that shares at least two properties with the target object, no objects that share only one property are returned. The specifics of these filters and the detailed rationale for them, as well as the exact formulas for computing similarity are given below.

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		x		x	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

Figure 4-3: Select-NN: Using properties of "`newspaper`," its nearest neighbors (the most similar objects) are identified. Up to ten are used similar objects are returned; here only two ("`book`" and "`magazine`") are shown.

The pseudocode in Figure 4-5 presents Select-NN formally. The pseudocode refers to a two-argument predicate *WNisa*, which is defined as follows:

The subsumption relationship "WordNet is-a" holds between objects O_1 and O_2 (denoted $WNisa(O_1, O_2)$) if there is a chain of "is-a" assertions in WordNet that leads from first WordNet sense of O_1 to the first or second WordNet senses of O_2. This definition has been selected empirically. Section 4.3.1 describes WordNet, WordNet senses, and the motivation for introducing *WNisa* and associated with it filtering in greater detail.

For ease of reference, all the steps of Select-NN are illustrated together in Figure 4-4. Computation of similarity in Select-NN in based on a generalization of Tversky's

contrast model of similarity. The model is formulated as:

$$Sim_{Tversky}(O, O') = \theta f(\mathcal{F}_O \cap \mathcal{F}_{O'}) - \alpha f(\mathcal{F}_O \setminus \mathcal{F}_{O'}) - \beta f(\mathcal{F}_{O'} \setminus \mathcal{F}_O)$$

where $\mathcal{F}_O$ represents the set of features that an object O possesses, $\mathcal{F}_O \setminus \mathcal{F}_{O'}$ denotes the set difference of the sets $\mathcal{F}_O$ and $\mathcal{F}_{O'}$ (i.e., features of O that O' does not have). The function f is typically assumed to be additive (simply returning the size of the set to which it is applied), and θ, α, and β are non-negative weights. Section 4.4 presents more details about adapting this model to domains in which some features are unknown, as well as incorporating into the model of some information-theoretic observations about semantic similarity made independently by Resnik and by Lin (Resnik, 1995, 1999; Lin, 1998).

The exact formulas used in Select-NN are as follows. Recall that $Tv(O, P)$ stands for the truth value of the assertion $A(O, P)$; "is true" assertions have $Tv = 1$ and "is false" assertions have $Tv = 0$.

Let $\mathcal{O}_P$ be the set $\{O : Tv(O, P) = 1\}$, i.e., the set of all objects O for which the property P has been asserted to be true, and $\|\mathcal{O}_P\|$ be the number of such objects. Then the *frequency weight* of P, denoted $FreqWt(P)$, can be defined as follows:

$$FreqWt(P) = \begin{cases} 2 & \text{if } \mathcal{O}_P = \emptyset, \\ 1 + 1/\log_2(\|\mathcal{O}_P\| + 1) & \text{otherwise.} \end{cases} \tag{4.1}$$

Note that the $FreqWt$ ranges between 1 and 2, $FreqWt(P) \in [1, 2]$, with larger values corresponding to properties that are true of fewer objects. The inverse of the logarithm is taken to make $FreqWt$ decrease as $\|\mathcal{O}_P\|$ increases. The motivation for assigning lower weight to more common shared properties is that, other things being equal, two objects sharing a very rare property are probably more similar than two objects sharing a very common one.[2] Giving greater weight to the rare features is consistent to prior approaches to measuring semantic similarity (Resnik, 1995; Lin, 1998). The motivation for $FreqWt$ is further discussed in Section 4.4 in light of the

[2] $\|\mathcal{O}_P\|$ is incremented by one to avoid division by zero (from $1/\log_2(1)$) when $\|\mathcal{O}_P\|$ is 1.

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		**x**		**x**	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

(a) Preparing to formulate questions about "`newspaper.`" Properties already known about "`newspaper`" are identified.

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		**x**		**x**	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

(b) All known properties of "`newspaper`" are used to look for similar objects.

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		**x**		**x**	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

(c) The most similar objects are identified (up to ten are used).

Figure 4-4: **Select-NN**: All steps together for the example of "`newspaper.`" For clarity, only "is true" properties are shown, marked with 'x'. Blank cells mean that nothing has been asserted.

SELECT-NN:
// $\exists A(O, P)$ denotes "O has property P" or "O does not have property P"
// has been asserted
// initialize scores to 0:
forall O_{src} **do**
  $Score(O_{src}) \leftarrow 0$;
end
forall $P : \exists A(O_{target}, P)$ **do**
  forall $O_{src} : \exists A(O_{src}, P), O_{src} \neq O_{target}$ **do**
    $Score(O_{src}) \leftarrow Score(O_{src}) + Wt(Tv(O_{target}, P), Tv(O_{src}, P), P)$;
  end
end
// If WordNet category is known, disallow objects with non-overlapping
// categories
// Always disallow objects subsuming O_{target}
// (see Section 4.2.3)
if $WordNetCateg(O_{target}) = \emptyset$ **then**
  let Candidate set $\mathcal{C} \leftarrow \{O : Score(O) > 0 \wedge \neg WNisa(O_{target}, O)\}$;
else
  let $\mathcal{C} \leftarrow \{O : Score(O) > 0 \wedge \neg WNisa(O_{target}, O)$
    $\wedge\ WordNetCateg(O) \cap WordNetCateg(O_{target}) \neq \emptyset\}$;
let $\mathcal{C}_2 \leftarrow \{O : O \in \mathcal{C} \wedge Score(O) \geq 2\}$;
// return only highest scoring objects
if $\|\mathcal{C}_2\| < 2$ **then**
  if $\|\mathcal{C}\| < 10$ **then**
    return $\mathcal{C}$ (with scores);
  else
    //return up to ten highest-scoring objects
    return $\mathcal{O}$ (with scores) : $(\|\mathcal{O}\| = 10 \ \wedge\ \forall O_{src}, O_{\neg src} : O_{src} \in \mathcal{O}, O_{\neg src} \notin \mathcal{O},$
      $Score(O_{src}) \geq Score(O_{\neg src}))$;
else
  if $\|\mathcal{C}_2\| < 10$ **then**
    return $\mathcal{C}_2$ (with scores);
  else
    return $\mathcal{O}$ (with scores) : $(\|\mathcal{O}\| = 10 \ \wedge\ \forall O_{src}, O_{\neg src} : O_{src} \in \mathcal{O}, O_{\neg src} \notin \mathcal{O},$
      $Score(O_{src}) \geq Score(O_{\neg src}))$;

Figure 4-5: Select-NN: Algorithm for selecting nearest neighbors.

contrast model of similarity and other prior work.

The amount by which the score of each object is updated is given by the function

$$Wt(Tv(O_{target}, P), Tv(O_{src}, P), P),$$

computed as follows:

$$Wt(Tv_1, Tv_2, P) = \begin{cases} FreqWt(P) & \text{if } (Tv_1, Tv_2) = (1,1), \\ -1.5 & \text{if } (Tv_1, Tv_2) = (1,0) \text{ or } (0,1)\ , \\ 0 & \text{if } (Tv_1, Tv_2) = (0,0). \end{cases} \tag{4.2}$$

The "punishment for mismatch" weight applied when $(Tv_1, Tv_2) = (1,0)$ or $(0,1)$ has been selected to be -1.5 as the negative of the average of the extreme values of the "reward for match" values, which are the range of $FreqWt(P)$ (recall that $FreqWt(P) \in [1,2]$). The motivation for making the weight the same for the two cases $(Tv_1, Tv_2) = (1,0)$ and $(Tv_1, Tv_2) = (0,1)$ is discussed further in Section 4.4.

Because a property of both the source and the target object could have been asserted to be true (Tv of 1) or false (Tv of 0), several cases arise in Eq. 4.2. (Recall that assertions other than "is true" and "is false" are not used in computing the similarity score). When both O_{src} and O_{target} share a property P (both truth values are 1), their similarity is incremented by $FreqWt(P)$, a weight that ranges between 1 and 2, as is detailed below. When the objects mismatch on this property (i.e. Tv is 1 for one object and 0 for the other), then their similarity is reduced by a constant 1.5. Finally, when both Tvs are zero, the similarity is not adjusted, the rationale being that two objects not having a property does not necessarily indicate much about their similarity. For example, neither a "`newspaper`" nor "`ice`" "can sing," but that does not make them any more similar.

The total similarity score is computed for every object whose score was updated. After some filtering that removes already known questions and questions that are taxonomically inferable (the filtering is also detailed in Figure 4-5 and in Section 4.2.3),

up to ten highest-scoring objects are selected and returned together with their scores, as illustrated in Figure 4-3. The reason behind forming the set $\mathcal{C}_2$ in Select-NN and looking at its cardinality is to avoid returning very weakly similar objects when a sufficient number of more similar objects are available. The score threshold of 2 ensures that at least two properties are shared; the threshold of 2 on cardinality of $\mathcal{C}_2$ has been chosen empirically.

4.2.2 Map-Props

The output of Select-NN — the source objects O_{src_i} paired with scored indicating their respective similarity to O_{target} — are passed to **Map-Props**, which proceeds as follows (pseudocode for Map-Props is given in Figure 4-8 and the formulae involved are presented later on in the text).

First, for every source object, Map-Props retrieves all properties asserted about this object. Both "is true" and "is false" assertions are included. For the target object "`newspaper`," the set of source objects may include "`book`" and "`magazine`," as illustrated in Figure 4-6.

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...		**x**		**x**	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

Figure 4-6: **Map-Props**: All known properties of the similar objects are selected.

Next, each mentioned property is mapped back onto the target object, with the total score of the mapping depending on the similarity scores of the relevant source objects and the kind of property being mapped (`subj`, `obj`, or `pp`). Already known properties of the target object ("`newspaper`," in our case) are filtered out, as are

properties that "`newspaper`" can be expected to have by taxonomic reasoning. Figure 4-7 illustrates the mapping of properties not yet known about "`newspaper`" back onto "`newspaper.`"

Objects	Properties *(with simplified form)*					
	...	contains knowledge *contain knowledge*	has pages *have page*	is cold *be cold*	is for reading *be read*	...
⋮		⋮	⋮	⋮	⋮	
book	...	x	x		x	...
ice	...			x		...
newspaper	...	**???**	**x**		**x**	...
magazine	...	x	x		x	...
⋮		⋮	⋮	⋮	⋮	

Figure 4-7: **Map-Props**: Properties (such as "`contains knowledge`") known about similar topics but not known about "`newspaper`" are formulated as questions about "`newspaper`" for presenting to contributors.

The pseudocode for the Map-Props is presented in Figure 4-8. The specific formulae used in calculating the scores of the mapped properties are as follows. For every source object, the score of every property known about it is updated, according to the product of three weights:

$$TvWt(Tv(O,P)) \times PropClassWt(PropClass(P)) \times SimWt(SimSc(O))$$

The component weight functions are computed as follows:

$$TvWt(Tv) = \begin{cases} 1 & \text{if } Tv = 1, \\ -1 & \text{if } Tv = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

$$PropClassWt(PropClass) = \begin{cases} 1.1 & \text{if } PropClass \text{ is } \texttt{subj}, \\ 1 & \text{if } PropClass \text{ is } \texttt{obj}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

MAP-PROPS:
// $\mathcal{O}_{src}$ denotes the set of objects given as input
// $SimSc(O)$ denotes the similarity score of an object O (to O_{target})
// $PropClass(P)$ denotes the property class of P (one of `subj`, `obj`, `pp`)
forall $O_{src} : O_{src} \in \mathcal{O}_{src}$ **do**
  forall $P : \exists A(O_{src}, P)$ **do**
    $Score(P) \leftarrow Score(P) + SimWt(SimSc(O)) \cdot PropClassWt(PropClass(P))$
    $\cdot TvWt(Tv(O, P))$;
  end
end
// select 100 properties with highest scores for further filtering
let $\mathcal{P} \leftarrow \{P : Score(P) > 0 \wedge \neg\exists A(O_{target}, P)\}$
if $\|\mathcal{P}\| < 100$ **then**
  return $\mathcal{P}$ (with scores)
else
  return $\mathcal{P}'$ (with scores) : $\{\|\mathcal{P}'\| = 100 \wedge \forall P_i, P_j : P_i \in \mathcal{P}', P_j \notin \mathcal{P}',$
  $Score(P_i) \geq Score(P_j)\}$;

Figure 4-8: Map-Props: Algorithm for mapping properties from nearest neighbors onto O_{target}

and

$$SimWt(SimSc) = 1 + \frac{\ln(SimSc)}{4}, \tag{4.5}$$

where $SimSc$ is the "similarity score" of O_{src} to O_{target} as returned by Select-NN. The equation for $TvWt$ simply ensures that the votes "for" and "against" this property holding for the target object are summed with the correct sign.

Note from Eq. 4.4 that $PropClassWt$ gives slight preference to subject assertions over object assertions, and does not increment the score of `pp` assertions at all, largely because of difficulties that arise with prepositions when the noun phrase is altered.[3] Also, $PropClassWt$ ensures that preference is given to statements which are perceived to be "about" the current topic O_{target}.

The equation for $SimWt$ (Eq. 4.5) ensures that the most similar objects in the set (according to Select-NN) have the greatest impact on the overall scores. The function has been chosen chosen after trying (informally) several alternatives; it is

[3]For example, one normally says "sit *at* a desk," but "sit *on* a bed"; changing the noun in the prepositional phrase would in a number of cases require an additional mechanism to modify the preposition to agree with the new noun.

highly sublinear to prevent over-emphasizing any given object, and to produce a truly "cumulative" analogy.

Once the total scores for each property are computed, properties already asserted about O_{target} are eliminated, and up to 100 of the highest scoring remaining properties are returned for further filtering. In practice, the computationally intensive work is not the computing of scores that has taken place up to this point; it is the linguistic processing of a sentence that happens at later stages. Thus it is important for performance reasons that this elimination happen at this stage and not later. The phenomenon of speed optimization causing propagation of some functionality from the tests into the generator is a commonplace feature of generate and test architectures. The properties with large negative scores (when such are present) are currently dropped, although they represent good hypothesis about what is *not* true about O_{target}.

4.2.3 Refinements in Select-NN

Refinements to Select-NN use taxonomic and semantic category information present in WordNet, a lexical database, as well as the taxonomic knowledge that LEARNER gathers. Extending LEARNER's taxonomy will override the information derived from WordNet. In this section, I briefly introduce WordNet's taxonomic handling of nouns and describe the implemented refinement to Select-NN which uses this information.

WordNet has approximately 80,000 noun word forms organized into approximately 60,000 *synsets* (for "synonym sets") (Miller, 1998). A concept may have multiple senses (e.g. "`table`" can mean a data table with rows and columns and a kind of furniture). The senses are numbered in order of decreasing frequency as they occurred in a manually tagged corpus, so that the first sense is the most common one in the corpus which was used to calculate the frequency data.

WordNet also provides taxonomic and other links ("is-part-of" and so on) between synsets. The taxonomic information states, for example, that the first sense of "`cat`" is a kind of the first sense of "`animal.`" See Miller (1998) for more information on how WordNet organizes nouns and their senses.

Furthermore, WordNet organizes all senses of nouns into high level categories. For example, for the two senses of "table" mentioned above, the table with rows and columns is in the category "group," and table that is furniture is in the category artifact.

In all, there are 26 such categories: act, animal, artifact, attribute, body, cognition, communication, event, feeling, food, group, location, motive, object, person, phenomenon, plant, possession, process, quantity, relation, shape, state, substance, time, tops. The tops category contains the concepts at the top of the hierarchy.

The refinement relating to WordNet categories (as detailed in Figure 4-5) is meant to remedy occasional generation of strange similarities when the system has insufficient knowledge. For example, two very different objects, such as a "mechanic" and "oil" can be similar to each other if little is known about each. If all that is known about both "mechanic" and "oil" is that both "can lubricate something," they will be judged to be similar by the system prior to WordNet filtering. The filtering based on WordNet categories will exclude "oil" from the set of objects similar to a "mechanic", because WordNet categories of senses of the noun "mechanic" do not intersect with any of the WordNet categories of senses of the noun "oil".

More specifically, to compute the set of WordNet categories for a topic O, the sense categories of the three most common WordNet senses of O are looked up (or, if there are fewer than three senses, categories of all the senses). For example, the noun "mechanic" has the single category person, and the noun "oil" has categories artifact and substance.

Another filter is the filter of taxonomic parents. It is implemented as follows. Recall the definition of the predicate *WNisa*:

The subsumption relationship "WordNet is-a" holds between objects O_1 and O_2 (denoted $WNisa(O_1, O_2)$), if there is a chain of "is-a" assertions in WordNet that leads from first WordNet sense of O_1 to the first or second WordNet senses of O_2.

Additionally, such statements as "a cat is a pet," "cats are pets," and "a cat is not a bird" in LEARNER's knowledge base are recognized as expressing presence and

absence taxonomic relationships between the subject and the object of the assertion. I denote these extracted relationships *LearnerIsA* and *LearnerIsNotA*. Given a target object O_{tgt}, all objects O such that

$$LearnerIsA(O_{tgt}, O) \vee (\neg LearnerIsNotA(O_{tgt}, O) \wedge WNisa(O_{tgt}, O))$$

are removed from the output of Select-NN.

4.2.4 Questions posed by cumulative analogy improve with acquisition of more knowledge

Importantly, knowledge acquisition by cumulative analogy exhibits *bootstrapping* qualities. The replies to the knowledge acquisition questions formulated by analogy are immediately added to the knowledge base, affecting the measure of similarity. If Select-NN incorrectly rates a non-similar object as too similar, many knowledge acquisition questions posed with the contribution of this object are likely to be answered "no," decreasing its similarity score in the future.

Even answering questions affirmatively is likely to strengthen the similarity scores of topics that are more similar, while leaving scores of other topics in the set of similar topics unchanged. This process is also likely to improve the quality of future questions. Here is an example: when starting with the seed knowledge base and teaching about "`newspapers`," the similar topics, together with their similarity scores, are: "`book`" (6.72), "`map`" (2.76) "`magazine`" (2.67), and "`bag`" (2.51). The three highest-scoring knowledge acquisition questions posed are "newspapers contain information?", "all newspapers have pages?" and "newspapers are for reading?" If these questions are answered affirmatively and the answers are submitted to the system, set of the similar objects remains the same, but their scores become: "`book`" (10.94), "`map`" (5.53) "`magazine`" (4.12), and "`bag`" (2.51). As can be seen from the change in similarity scores, the less similar topic ("bag") became less influential in creating knowledge acquisition questions relative to others. This should lead to questions posed by LEARNER being more focused.

Conversely, as more "is true" assertions are added, similar objects that also share those properties but were not previously in the top ten most similar object will join that group.

4.2.5 Other merits of cumulative analogy

In this section, I discuss some important merits of the concept of cumulative analogy. The specific algorithmic implementation of cumulative analogy and some of its shortcomings are addressed following a detailed evaluation, in Section 8.5.1. My approach to measuring similarity is motivated and compared to other approaches in the literature in Section 4.4.

First, cumulative analogy is *noise tolerant* at all stages. Select-NN sums evidence for similarity from many individual properties, limiting the effects of spurious matches. Map-Props then sums evidence for each property from up to ten similar objects, further limiting the effect of any residual noise in Select-NN's output. Noise tolerance is of particular importance when the system is forced to pose questions when it knows very little about the topic at hand.

An example of cumulative analogy exhibiting noise tolerance in the step of creating a set of knowledge acquisition questions from the similar objects returned by Select-NN is as follows. Consider the similar objects to the object "`tool`" in the seed knowledge base, presented in Table 4.1.

computer	7.61
machine	5.39
horseshoe	5.26
fire	3.92
knife	3.90
car	3.90
wrench	3.76
musical instrument	3.06
fan	2.71
weapon	2.45

Table 4.1: Objects similar to "`tool`" in the seed knowledge base.

Arguably, the similar topic "`fire`" (and perhaps the similar topic "`horseshoe`")

are spurious. "Tool" was judged to be similar to "fire" because of the following pairs of assertions: "Humans use tools/fire", "A tool can help a person/Fire can help people", and "Tools are useful/Fires can be useful". Despite these matches, the top five questions posed about "tools" as follows (shown together with the similar topics from which they are mapped):

- tools can run on electricity? (computer, machine, fan)
- tools are machines? (computer, car, fan)
- a tool can hurt a person? (fire, car, weapon)
- tools are man-made? (computer, machine)
- tools are complicated? (computer, car)

In this particular case, only the third question was mapped with participation of "fire," and this question had additional support from the objects "car" and "weapon." Other, irrelevant properties of "fire" present in the knowledge base (such as "a fire is hot", "fire consumes oxygen", "fire can burn a house") are not posed as knowledge acquisition questions.

I refer to to this ability of cumulative analogy to focus in on the more relevant properties under the general term "noise tolerance." It is important that the "noise" that cumulative analogy tolerates has many sources. One source is spurious matches that arise from insufficient knowledge (for example, a "tool" and a "fire" were similar because of such assertions as "Humans use tools/fire" and no knowledge that would state how a "tool" is different from a "fire" was present. Another source is lexical ambiguity, which may cause semantically different assertions to be judged to be similar by LEARNER (as discussed further in Chapter 6). Finally, another source of noise may be some unintentional or malicious misinformation of the system by a contributor. As long as the total amount of "noise" from all of the possible sources does not overwhelm cumulative analogy's ability to tolerate noise, cumulative analogy will still pose reasonable questions even when forced to operate over such "noisy" data.

Second, the knowledge collected by LEARNER tends to be *syntactically uniform.* By virtue of the algorithm (contributors are encouraged to confirm, deny, or correct statements mapped from other similar statements), syntactic variation is kept low, simplifying further processing and enabling the analogy mechanism to work well on consequent iterations.

Finally, the approach conforms to the guiding principle of transparency put forth in Section 4.1. Contributors can immediately see the impact of their contribution on the system's beliefs about similarity of objects. The impact is shown by presenting the current beliefs about similarity, the pairs of assertions that support the current beliefs about the similarity, and the changes in the similarity caused by the knowledge just added by the contributor are all presented or made available to the contributor, as detailed in Section 5.1.

4.3 Filtering questions with critics

Generation of knowledge acquisition questions in LEARNER consists of two components: the generation component and the filter, or critic component. Critics are the part of the architecture that implements some commonsense requirements on the knowledge acquisition questions. A good set of critics offloads this functionality from the generators, allowing the generators to be simpler and easier to implement. Also, because critics implement commonsense requirements on questions, they are more fixed in their nature than generators, which have a much broader charter of "generating plausible assertions" (to be converted into questions).

In the implemented system, critics receive and process assertions consisting of the object O_{target}, some property P and the assertion's score as computed by Map-Props. These assertions are later converted into natural language. Although in the envisioned architecture critics are allowed to combine and modify their inputs, the implemented critics do not modify the assertions or their scores. They are only allowed to "veto" (filter) some of them, passing the filtered set to the natural language generation component and on to the interface.

The implemented critics embody the following principles:

- Do not ask what you already know.
- Do not ask things that can be inferred with high confidence (only taxonomic inference has been implemented).
- Do not ask what you can not understand (do not pose non-parsable questions).

Several of the above principles can be seen as instantiations of an even more general information-theoretic principle that the system should pose the questions that will yield the most information (see Chklovski (1998) for an example of applying this principle to selecting a recognition "question" in a simplified object recognition task). The implementation of the principle "do not ask what is already known" is straightforward, as is implementation of "do not ask what you can not understand" (the new sentences that cannot be parsed are filtered out). The next section details filtering out assertions inferable with a simple inference mechanism.

4.3.1 Using a taxonomy to filter questions

The core question-posing algorithm operates on correlations in the supplied knowledge base. It requires no additional sources of information, and does not have a deep understanding of what the natural language assertions mean or how they relate to each other semantically.

Such a lightweight approach has the advantage of simplicity. However, it has some limitations. Namely, if a property holds for many taxonomically close objects, the algorithm will never really "get it," and will pose the same question about many objects. For example, when told that "animals have body parts," the generation component may, in generating questions for various topics, ask whether "monkeys have body parts," "cats have body parts," "dogs have body parts," and so on.

In this section, I present an implemented critic that adds awareness of taxonomic relationships and taxonomic inferences, and includes handling of quantifiers such as "some."

This filter is used only on subject-assertions. Other assertions are permitted by this critic without evaluation. The reason is that in interpreting natural language statements, the syntactic subjects of the sentences are by default scoped universally, while syntactic objects are not. For a discussion of quantifiers in first-order predicate calculus expressions generated by interpreting natural language assertions, see Jurafsky and Martin (2000, p. 558) and Alshawi, Carter, van Eijck, Gambäck, Moore, Moran, Pereira, Pulman, Rayner and Smith (1992). The sources present examples of syntactically identical sentences that have different quantifier assignments when interpreted semantically. The second source describes a set of heuristics that in some (but not all) cases make the correct quantifier choices. For example, in the sentence "people have body parts" the subject "`people`" is universally quantified, while the object "`body parts`" is not (that is, the statement does not assert that people have every possible body part, such as wings).

Recall the two-argument predicate $WNisa(O_1, O_2)$ described in Section 4.2.3. The predicate computes whether O_1 is a kind of O_2 according to the taxonomic information about word senses in WordNet.

The taxonomy-based critic of assertions works as follows. For each property P returned by Map-Props, the critic evaluates whether to filter the assertion $A(O_{target}, P)$. The critic looks for the maximally specific generalization of O_{target}, O_{sup}, such that truth or falsehood of P is asserted about O_{sup}. That is, the critic looks for the most specific O_{sup} such that $WNisa(O_{target}, O_{sup})$ and $\exists A(O_{sup}, P)$. If no such O_{sup} exists, $A(O_{target}, P)$ is not filtered. Otherwise, the natural language sentence that gave rise to $A(O_{sup}, P)$ is analyzed. The assertion is not filtered if the subject noun phrase is quantified with any of the determiners "most," "many," "some," "several," or "few." Otherwise, $A(O_{target}, P)$ *is* filtered, and no question about O_{target} having the property P will be posed.

To work in conjunction with this critic, Select-NN also includes a step that removes objects subsuming O_{target} from the set of objects it returns. The motivation for this filter is the later filtering of *properties* that can be computed to be inherited from a more general object, described in Section 4.3.1. If Select-NN did not filter ob-

jects subsuming O_{target}, consequent application of Map-Props to these objects would have generated many questions that would be immediately filtered out again by the taxonomic critic.

4.4 Measuring semantic similarity

In this section, I place the computations of similarity used in LEARNER in the context of research on human judgments of semantic similarity and discuss prior approaches to generating machine semantic similarity judgments. In addition to this discussion, Section 7.5 further expands my discussion of semantic similarity. Additional topics include linguistic and world phenomena that give rise to semantic similarity, the importance of dissimilarity between similar categories, and human bias in selection of features that define similarity (related to the theorem of the Ugly Duckling, which states that in the absolute absence of bias any two categories of objects are equally similar).

Assessment of semantic similarity is essential to a variety of NLP tasks (Montemagni and Pirelli, 1998). Typically, the notion of similarity is approached via the notion of distance. Introducing a way to measure distance between any two objects allows identification of objects that are near each other as the most similar.

The intuition of a well-behaved distance measure is captured in a notion of a *metric*. In order for distance measure D, defined over a set of objects $\mathcal{O}$ to be a metric, the following conditions must hold for any three objects O, O' and O'' in $\mathcal{O}$:

non-negativity: $D(O, O') \geq 0$,
reflexivity: $D(O, O') = 0$ if and only if $O = O'$,
symmetry: $D(O, O') = D(O', O)$,
triangle inequality: $D(O, O') + D(O', O'') \geq D(O, O'')$.

Assuming that objects are represented as vectors of real-valued or binary properties (and assuming the value of each property is known for each object), objects can

be thought of as points in an N dimensional space, where N is the number of distinct properties. In feature-based approaches to similarity, a distance metric between objects represented as feature vectors is often defined using a specific value of $r(r > 0)$ of the Minkowski metric:

$$L_r(O, O') = \left(\sum_{k=1}^{N} |O_k - O'_k|^r \right)^{\frac{1}{r}}$$

where O_k is the value of the k^{th} feature of O (Duda, Hart and Stork, 2000; Goldstone, 1999). L_1 (i.e., the above expression for $r = 1$) is known as *Manhattan* or *city-block* distance and is closely related to *Spearman's footrule*. L_2 is the familiar Euclidean distance. The similarity is then defined to be inversely related to the distance measure. L_1 and L_2 are the most commonly used in feature-based similarity measures (Goldstone, 1999).

On the other hand, studies of human similarity judgments indicate that when humans estimate pairwise semantic similarity (or "distance") of concepts, they systematically violate all of the above properties of distance metrics save, perhaps, for non-negativity (Tversky, 1977; Goldstone, 1999).

Human judgments violate reflexivity because not all pairs of identical objects are judged to be equally similar. For example, complex identical objects (such as identical twins) have been empirically observed to be judged more similar than simple objects (such as squares). Symmetry is violated in the following way: when an object with many features is compared to one with fewer features, the one with more features is judged to be less similar to the other one than vice versa. For example, subjects in the United States judged New York to be less similar to Tel Aviv than vice versa and China to be less similar to North Korea than North Korea is to China (Tversky, 1977; Goldstone, 1999). Finally, triangle inequality can be violated when O and O' and O' and O'' are similar because of different sets of features, and O and O'' have little in common. For example, "ball" (O) and "moon" (O') are both round, and "moon" (O') and "lamp" (O'') both can give off light, but "ball" and "lamp" are less similar than either of the above pairs (Tversky and Gati, 1982; Goldstone, 1999).

In choosing measures of distance, there is usually little agreement on a principled way of selecting one (even if restricted only to metric measures). For example, Aggarwal et al. state: "In most high dimensional applications the choice of the distance metric is not obvious; and the notion for the calculation of similarity is very heuristical" (Aggarwal, Hinneburg and Keim, 2001).

In light of these observations, I have chosen not to build LEARNER's measure of similarity on a notion of metric distance, but instead to adopt Tversky's contrast model of similarity. The contrast model is formulated as:

$$Sim_{Tversky}(O, O') = \theta f(\mathcal{F}_O \cap \mathcal{F}_{O'}) - \alpha f(\mathcal{F}_O \setminus \mathcal{F}_{O'}) - \beta f(\mathcal{F}_{O'} \setminus \mathcal{F}_O)$$

where $\mathcal{F}_O$ represents the set of features that an object O possesses, $\mathcal{F}_O \setminus \mathcal{F}_{O'}$ denotes the set difference of the sets $\mathcal{F}_O$ and $\mathcal{F}_{O'}$ (i.e., features of O that O' does not have). The function f is typically assumed to be additive (simply returning the size of the set to which it is applied), and θ, α, and β are non-negative weights.

The contrast model is typically applied to domains in which perfect information is assumed — for a given object any given feature is known to be either present or absent. On the other hand, by the nature of the task of knowledge acquisition, LEARNER manipulates features whose (binary) values are sometimes unknown. In generalizing the contrast model to domains with unknown features, I use a slightly different notation, using $\mathcal{P}_O$ to denote the set of "is true" properties of O, and $\mathcal{P}_{\neg O}$ to denote the set of "is false" properties of O. I have also chosen a measure that, when measuring similarity between O and O', is not affected by features unknown about either O or O'. That is, in place of $\mathcal{F}_O \setminus \mathcal{F}_{O'}$, I use $\mathcal{P}_O \cap \mathcal{P}_{\neg O'}$, the set of properties (features) that are known to be true for O and are known to be false for O'. Similarly, I use $\mathcal{P}_{O'} \cap \mathcal{P}_{\neg O}$ in place of $\mathcal{F}_{O'} \setminus \mathcal{F}_O$.

My approach to measuring similarity also weights individual features differently, depending on their frequency, in the term $\mathcal{P}_O \cap \mathcal{P}_{O'}$ corresponding to $\mathcal{F}_O \cap \mathcal{F}_{O'}$.

Features in this term are weighted according to Eq. 4.1, reproduced here:

$$FreqWt(P) = \begin{cases} 2 & \text{if } \mathcal{O}_P = \emptyset, \\ 1 + 1/\log_2(\|\mathcal{O}_P\| + 1) & \text{otherwise.} \end{cases}$$

where $\mathcal{O}_P$ is the set of objects for which the property P is asserted to be true. This weighting gives more weight to the features that are rare, similar to the approach taken by (Resnik, 1995; Lin, 1998). Additionally, this weighting ranges between 1 and 2, $FreqWt(P) \in [1, 2]$, to keep the contribution of any one feature from dominating the overall sum and thus remain consistent with the spirit of Tversky's contrast model. Operating on the logarithm of the number of features in $\mathcal{O}_P$ rather than on the number of features in $\mathcal{O}_P$ itself, $\|\mathcal{O}_P\|$, is motivated by two considerations: (i) as will be presented below, models proposed by other researchers propose measures that operate on logarithms of frequencies based on information-theoretic considerations (Resnik, 1995; Lin, 1998); indeed, Resnik points out that this may be an important commonality of the models (Resnik, 1995, p. 4), and (ii) by taking the inverse of the logarithm, *FreqWt* assumes values that are more uniformly distributed along the interval $[1, 2]$ for a larger range of values of $\|\mathcal{O}_P\|$.

For properties that are "is true" about one object being compared and are "is false" of the other (i.e., $P \in \mathcal{P}_O \cap \mathcal{P}_{\neg O'}$ or $P \in \mathcal{P}_{O'} \cap \mathcal{P}_{\neg O}$), giving greater weight to properties that are rare does not seem motivated. After all, having opposite truth values on a rare property should not have greater impact than disagreeing on a more common one. In light of this, in my model equal weight is given to all properties in sets $\mathcal{P}_O \cap \mathcal{P}_{\neg O'}$ and $\mathcal{P}_{O'} \cap \mathcal{P}_{\neg O}$. The magnitude of the negative weight was chosen to be the average of the extreme values that the positive contributions can assume, i.e, 1.5. The resulting measure of similarity used in this work is:

$$Sim(O, O') = \left(\sum_{P \in \mathcal{P}_O \cap \mathcal{P}_{O'}} FreqWt(P) \right) - 1.5 \times \|\mathcal{P}_{O'} \cap \mathcal{P}_{\neg O}\| - 1.5 \times \|\mathcal{P}_O \cap \mathcal{P}_{\neg O'}\|$$

Because I have selected $\alpha = \beta$ (both are equal to 1.5), symmetry of distances is

preserved in my adaptation of the contrast model, i.e., for any two objects O and O', $Sim(O, O') = Sim(O', O)$. Note, however, that LEARNER's algorithms are affected only by relative values of similarity to the set of most similar objects being considered. For example, the measure

$$Sim'(O, O') = \frac{Sim(O, O')}{Sim(O, O)}$$

is equivalent to the measure $Sim(O, O')$ in terms of the knowledge acquisition questions it yields, but $Sim'(O, O')$ is not symmetric (i.e., $Sim'(O, O') \neq Sim'(O', O)$), with $Sim'(O, O') < Sim'(O', O)$ when more is known about O than O', as in the above examples cited from Tversky.

Note that regardless of whether Sim or Sim' is used, the triangle inequality can be violated, and hence neither Sim nor Sim' meets the requirements of being a metric. Because similarity between any two objects can still be computed, the notion of "nearest neighbors" is still well-defined. However, I caution the reader that "nearest" in "nearest neighbors" does not refer to a distance in a metric space.

For completeness, at this point I overview some other approaches to measuring similarity that have appeared in the literature. A derivation of a metric measure of similarity from a set of assumptions about its desired properties can be found in (Lin, 1998). This information-theory motivated approach can be applied to measure similarity in many settings. When applying it to feature-based semantic similarity judgments between words, Lin formulates it as:

$$Sim_{Lin}(O, O') = \frac{2 \times I(\mathcal{F}_O \cap \mathcal{F}_{O'})}{I(\mathcal{F}_O) + I(\mathcal{F}_{O'})} \tag{4.6}$$

where $I(\mathcal{F})$ is the amount of information contained in a set of features $\mathcal{F}$. Under the assumption of independence of features, $I(\mathcal{F}) = -\sum_{f \in \mathcal{F}} \log P(f)$, where $P(f)$ is the probability of feature f (Lin, 1998). The probability $P(f)$, in turn, can be estimated from frequency counts of encountering the feature f in the knowledge base.

Another feature-based distance metric is the Tanimoto metric, as presented, for

example, in (Duda et al., 2000, p. 188,541):

$$D_{Tanimoto}(O, O') = \frac{\|\mathcal{F}_O\| + \|\mathcal{F}_{O'}\| - 2\|\mathcal{F}_O \cap \mathcal{F}_{O'}\|}{\|\mathcal{F}_O\| + \|\mathcal{F}_{O'}\| - \|\mathcal{F}_O \cap \mathcal{F}_{O'}\|} \tag{4.7}$$

which is normalized by the number of features and has the range $[0, 1]$. As can easily be established by constructing a Venn diagram for $\mathcal{F}_O$ and $\mathcal{F}_{O'}$, Tanimoto distance can also be written as:

$$D_{Tanimoto}(O, O') = 1 - \frac{\|\mathcal{F}_O \cap \mathcal{F}_{O'}\|}{\|\mathcal{F}_O \cup \mathcal{F}_{O'}\|} \tag{4.8}$$

Note that low distances correspond to high similarity and vice versa. Also note that if values of some features are unknown, applying this formula to only the known features can lead to violation of the triangle inequality (recall the example above involving "ball," "moon," and "lamp"). As stated, neither the Lin nor the Tanimoto approach to measuring similarity distinguishes between the cases where a feature is present in O but is either absent or unknown about O'.

Additional prior work introduces similarity metrics based not on comparing features of objects, but on taxonomic and corpus frequency information. Lin's general derivation can be instantiated for taxonomy-based similarity as:

$$Sim_{Lin-tax}(O, O') = \frac{2 \times \log P(C_{O,O'})}{\log P(C_O) + \log P(C_{O'})} \tag{4.9}$$

where C_O is the most specific category containing O, $C_{O'}$ is the most specific category containing O', and $C_{O,O'}$ is the most specific category containing both O and O' (assuming the taxonomy is a tree) (Lin, 1998). $P(C)$ denotes the probability of encountering a concept from category C (including any concept from a category subsumed by C), which can be estimated from, for example, a textual corpus.

To measure similarity of concepts organized in a multiple inheritance hierarchy Resnik has used the following distance metric:

$$D_{Resnik}(O, O') = \max_{C_{O,O'}}[-\log P(C_{O,O'})] \tag{4.10}$$

where $C_{O,O'}$ ranges over the set of categories that subsume both O and O', and $P(C)$ is the probability of encountering a reference to the concept O in a corpus (Resnik, 1995). Resnik reports encouraging results for the ability of this measure to replicate human judgments (with the correlation r = 0.79, with a benchmark upper bound of r = 0.90 of correlation of similarity judgments by different humans). The similarity measure proposed by Lin is reported to perform slightly better on the same test set (Lin, 1998). For further review and comparison of various taxonomy-based metrics as well as some results on their ability to replicate human similarity judgments, see (Resnik, 1999; Lin, 1998).

Property	**Learner (contrast model)**	**Lin**	**Resnik**	**Tanimoto**
Mimics qualitative features of human judgments	✓			
Accounts for differences	✓	*		
Feature-based (able to bootstrap)	✓	✓	*	
Is a metric measure		✓	✓	✓

Table 4.2: Summary of properties of the reviewed measures of similarity. The first and the last properties in the table are mutually exclusive. "Accounts for differences" row indicates whether presence of a feature with opposite truth values decreases similarity. For a feature known about an object, Lin's measure does not distinguish the features being unknown and known to be false about the other object. "Feature-based" row indicates whether the model works with features of objects rather than purely taxonomic information. Resnik's measure uses taxonomic position and corpus frequency. Measures based on features will tend to improve their accuracy as values of more features are acquired.

Overall, there are several properties that are desirable in a similarity measure. This discussion has focused on the following:

- the ability to mimic human judgments,
- the ability to account for differences between objects as well as for similarities,
- meeting the criteria for being a metric measure.

In the case of incomplete information and in knowledge acquisition scenarios, I believe an important additional feature is a measure's ability to improve as values of

more features become known about more objects. In Table 4.2, I summarize how the measures reviewed in this section meet the above desiderata. Note that in selecting a measure to use in the LEARNER, I considered the potential to mimic empirically observed human judgments more important than whether the measure is a metric. The measure actually used in LEARNER comes out ahead of others according to the specified evaluation criteria.

Chapter 5

Interface

In this chapter, I describe the interface, describe how it conforms to the guiding principles laid out previously, and describe the multiple-choice answers that LEARNER admits.

5.1 Interface description

To structure the elicitation, a topic-centric approach is used; that is, there is always an identified noun phrase that is the topic of the current acquisition. Given this requirement, there are two high-level issues: what topic to talk about, and what to ask about that topic.

The topic of acquisition is selected by the user. The system exerts only a slight influence on the selection of the topic, in the following way: when the system presents the knowledge acquisition questions, it also presents similar topics (the output of Select-NN); each of these can be clicked to become the new topic of acquisition. Refer to Figure 5-1 for an example.

Given a topic, the system takes a mixed-initiative approach to elicitation — the contributor is given a chance to select a topic and enter some assertions about it. Once the system has some knowledge about a topic, it transitions to active acquisition mode, using the present knowledge to formulate further knowledge acquisition questions. Because this "dialogue" is with a device with a display, it is somewhat

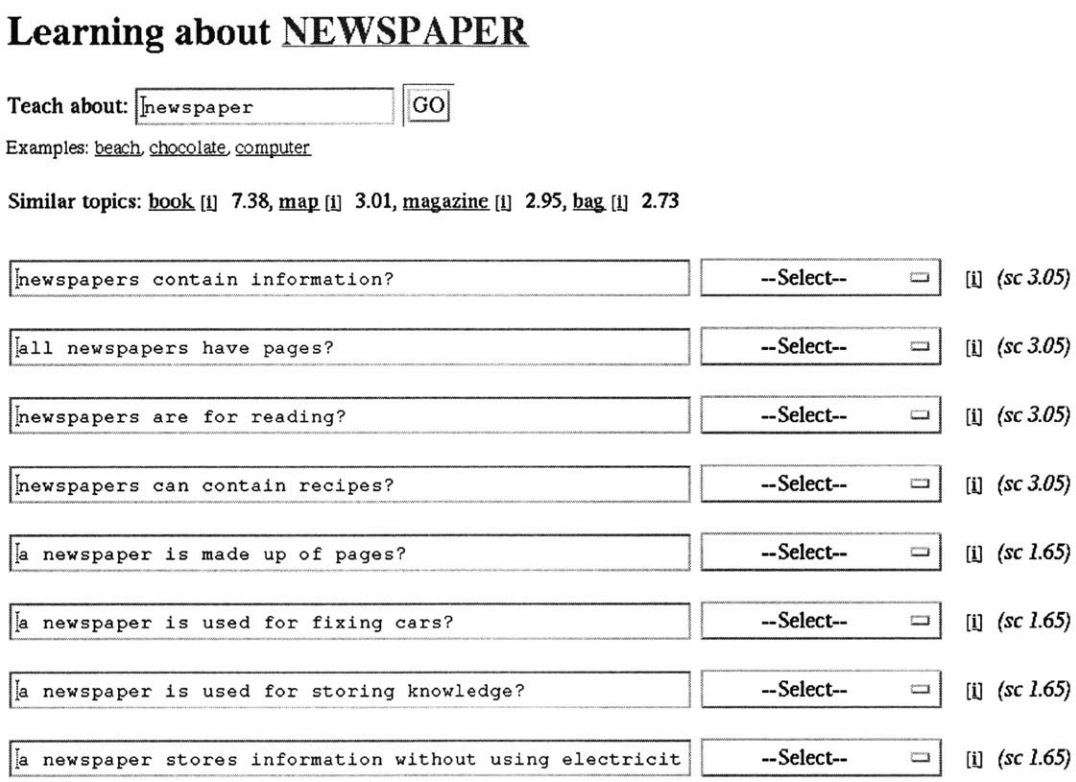


Figure 5-1: Screenshot of acquiring knowledge about "`newspaper.`" Reproduces Figure 2-1.

different from a verbal dialogue: the system's top N (currently 20) questions are presented in a batch, and the contributor reacts to or ignores each one, pressing a button when done with the batch. The opportunity to add additional assertions that are not responses to the system's questions is also present on every screen.

In addition to displaying the similar topics, the system adheres to the goal of being transparent by providing information about (i) how it has arrived at the similar topics (ii) how it has arrived at each particular question, and (iii) which additional questions were filtered in the process.

The interface provides an '[i]' ('i' for "info") hyperlink next to each similar topic. Clicking it shows how much each pair of matching signatures between O_{src} and O_{target} have contributed to the similarity score. Refer to Figure 5-2 for an example of the system presenting the reasons for similarity of "newspaper" and "book."

Similarity of 'Newspaper' and 'Book'

5 reasons affected similarity. Total score is **6.73**.

Score	Source Assertion	Similar Assertions
1.43	**Newspapers** are a source of information	**Books** are a source of information
1.36	**Newspapers** are printed on paper	**Books** are printed on paper
1.33	A **newspaper** can be read	**Books** are to be read **Books** can be read
1.32	People can read **newspapers** People read **newspapers** Some people read the **newspaper**	A person can read a **book** Most people can read **books** People can read **books** People read **books**
1.29	**Newspapers** are made of paper	**Books** are made of paper

Figure 5-2: Presenting to the contributor reasons for similarity of "`newspaper`" and "`book`".

The interface also provides an '[i]' hyperlink next to each knowledge acquisition question. Clicking it shows what source objects and assertions caused Map-Props to formulate this question. Refer to Figure 5-3 for an example.

The filtering of assertions inferable with taxonomic reasoning (as described in Section 4.3.1) is reported in gray directly before the presentation of the questions

Reason for asking 'Newspapers contain information?':

By analogy from these assertions about similar topics:

Book [i]	Books contain information
Map [i]	Maps contain information

Figure 5-3: Presenting to the contributor the reasons for formulating the question "newspapers contain information?" The question was formulated by analogy from similar topics "book" and "map."

that passed the filter. For example, the system may present that the assertion "cats have bones" was filtered like this:

"animals have bones" **prevents asking** "cats have bones."

The principle of transparency manifests itself in one further way when the system provides contributors with feedback about the effect of their contribution. When a contributor adds knowledge about a topic and requests new knowledge acquisition questions from the system, the system recomputes the set of similar topics, incorporating the new knowledge, and displays how each new assertion has affected the set of similar topics. For example, upon learning "newspapers contain information" the system may display:

Added "newspapers contain a information" (effect: ▲ book),

indicating that "`newspaper`" became more similar to "`book`" as a result of this addition.

The symbols '▲' (in green) and '▼' (in red) are also used when displaying the set of similar topics to indicate the direction of change in their similarity scores.

5.2 Permitted multiple-choice answers

To generate a knowledge acquisition question, a question mark is simply appended at the end of an assertion (with other trailing punctuation removed), and the assertion is presented as a "question." This question is presented in an HTML text input field, so

that the contributor can modify the question. (For example, rather than replying "no" to the question "cars run on steam," the contributor can alter it to read "cars run on gasoline," and assert that by answering "yes"). More often, however, the contributors leave the question unaltered, merely selecting one of the available predefined answers:

- Yes
- No
- Some / Sometimes
- Matter of opinion
- Nonsensical question

This set of allowed answers has been selected as a result of an analysis of the types of questions posed. Choosing the set of allowed answers is important because, once fixed, the users have no further control over these.

The main criterion used in selecting the set of answers was that the list should be short and should make it easy for the contributor to answer the questions. In other words, the answers should capture the options that come up.

Currently, only the answers "yes" and "no" affect the similarity scores of Select-NN and Map-Props. The other answers are simply stored with the assertions they correspond to, and the same question is not re-posed.

The motivation for each type of answer follows. "Yes" and "No" are clear enough. The need for the answer "some/sometimes" arises when a system overgeneralizes. Consider needing to answer the question "living things have gills" (some), or "traffic lights are yellow" (sometimes).

"Matter of opinion" is included to address questions such as: "donkeys are beautiful," "there is life after death." I include this option to steer users away from answering "yes" or "no" to questions that others may give a different answer to. This is intended to make the similarities drawn by Select-NN more acceptable to everyone.

The option "Nonsensical question" is present to help evaluate the performance of the system. To improve LEARNER further, knowing what fraction of questions are

nonsensical and what conditions give rise to nonsensical questions is important. See the discussion in Section 8.1 for additional discussion of "nonsense question" answer and the frequencies of each answer.

Finally, one answer type that was considered but was not added, is "I (personally) don't know." The need for such an answer may arise when the system poses a question that a contributor believes has an answer, but the contributor personally does not know it. Consider, for example, the assertion "Volga is the longest river in Russia." In my experience, (perhaps because the system focuses on "commonsense" knowledge), the need for such an answer is sufficiently rare. Currently, the contributor, not having a useful answer for LEARNER, may simply skip such questions. The question will then likely be posed to another contributor.

Chapter 6

Ambiguity

There is much ambiguity in language. For example, individual words are ambiguous: the word "mouse," even when used as a noun, can refer to both a kind of rodent or a kind of pointing device. Ambiguity exists at the word, phrase, and assertion level, and can interfere with using knowledge correctly.

In this chapter, I identify several kinds of ambiguity, discuss which approaches to processing knowledge are less and more sensitive to presence of ambiguity in the knowledge, and outline approaches to removing ambiguity, with emphasis on ambiguity present in the data LEARNER collects.

6.1 Kinds of ambiguity

There are many kinds of ambiguity in language one can identify (Jurafsky and Martin, 2000, pp. 372–376, 631–646). The kinds most relevant to LEARNER are as follows:

Word boundaries and base forms. In some languages, such as written Chinese, merely establishing word boundaries in a sentence can be a challenge because no special demarcation (such as a space) is present between symbols comprising different words. This problem does not normally arise in typed English text. However, another problem that sometimes does arise in English is establishing the base forms of words (to form the sentence signatures, for example). Consider

determining the base form of the word "putting": it may be either "put" or "putt." Admittedly, such cases are rare and LEARNER makes no effort to handle them correctly (it will always select "put" in the above example).

Word sense. Also known as *lexical ambiguity.* Consider the seemingly simple statement "many computers have a mouse." "Computers" are the computing devices, not people who perform calculations, "have" means "have as part" (as opposed to "have a meal" or "have a baby"), and "mouse" refers to a pointing device.

In addition to homographs (words of different origins having the same spelling), sometimes a single word can refer to two or more related, but distinct concepts. For example, consider the word "coffee": it can refer either to the drink made of coffee beans or the beans themselves. Only the drink is normally liquid, and it can be important to understand this in order to reason about the coffee beans.

Structural. Structural ambiguity arises when a sentence can be parsed in several different ways. Three common kinds of structural ambiguity are usually identified:

- Attachment ambiguity
- Coordination ambiguity
- Noun-phrase bracketing ambiguity

Attachment ambiguity refers to not knowing how pieces of the sentence fit together. It often arises with prepositional phrases, for example:

> People can sometimes see the Grand Canyon flying from LA to New York.

Syntactically, it is unclear whether "flying from LA to New York" modifies "people" or "the Grand Canyon."

Another common kind of parsing ambiguity is coordination ambiguity. It stems from an interaction of modifiers and conjunctions; it currently does not arise in

LEARNER because conjunctions are disallowed (see Section 3.2). I present an example for completeness. Consider:

"Young dogs and cats drink milk."

Two interpretations are possible:

"Young dogs and young cats drink milk," or
"Cats and young dogs drink milk."

Noun-phrase bracketing ambiguity stems from bareness of form of noun phrases. For example, "complete peace plan" can be interpreted as a "plan" for "complete peace," or a "complete" "peace plan."

There are additional, more exotic kinds of structural ambiguity, often arising from uncertainty about the part of speech of a certain words. Consider:

"Fruit flies like a banana,"

which can mean either "(fruit flies)$_{subj}$ like$_{verb}$ a banana," or "fruit$_{subj}$ flies$_{verb}$ like a banana."

Currently, LEARNER collects assertions and uses them in further knowledge acquisition without taking any specific steps to remove any of these types of ambiguity. Given the ambiguity present in the seed knowledge base and in the knowledge being collected, there are two issues to address: (i) how the ambiguity in the knowledge affects the acquisition algorithm, and (ii) how the ambiguity in the knowledge affects the usefulness of the collected knowledge for other efforts and how the ambiguity in the knowledge may be reduced or eliminated.

The next section discusses the impact of ambiguity on the cumulative analogy algorithm, and the following two sections respectively, discuss which possible uses of the knowledge base are and are not likely to be hampered by the ambiguity in the knowledge base and how the ambiguity may be ameliorated.

6.2 Lexical ambiguity: impact on knowledge acquisition

The assertions collected by LEARNER are quite simple syntactically. Conjunctions or disjunctions are not allowed, and complex sentence structure that could lead to structural ambiguity, such as attachment ambiguity, is rare. In contrast, the assertions often rely on frequently used words, which tend to have high polysemy counts (that is, have large numbers of meanings).

Motivated by this observation, I focus on the lexical ambiguity and its effect on the acquisition algorithm. My analysis is structured according to an observation about the places where lexical ambiguity can arise in applying cumulative analogy.

Lexical ambiguity can arise in four different places in the process of applying cumulative analogy:

- Ambiguous target topic of acquisition O_{target},
- Ambiguous source topic O_{src},
- Ambiguous word in a property P being used to calculate similarity of O_{target} and O_{src}, and
- Ambiguous word in a property P being mapped from O_{src} onto O_{target}.

The impact of each of the four conditions on the quality of questions posed by cumulative analogy is examined in turn.

Ambiguous O_{target}. Let us consider an example of acquiring knowledge about an ambiguous topic O_{target}, namely "shower." It has two senses, one similar to "rainfall" and another similar to "bathtub." Assertions about both senses of "shower" will be used in finding near neighbors, and as a result Select-NN may retrieve some near neighbors for each sense.

This selection of near neighbors for more than one sense may impact the acquisition in two ways. One is that fewer near neighbors will be used per sense,

therefore limiting the benefit of the noise-canceling ("cumulative") quality of the algorithm. This may lead to questions with less support, and thus perhaps of lower quality, to be posed. The second way is that questions posed may be about different senses of the topic, causing some confusion in the contributor who has to reinterpret which sense of topic is being used from question to question. In the worst case, some assertions may be interpretable for either sense of the topic, but have different truth values depending on the sense. In such cases, the collected knowledge needs to be disambiguated (using the methods discussed in Section 6.4) before the collected truth value becomes useful.

Ambiguous O_{src}. When one or more of the source topics O_{src} (the nearest neighbors) returned by Select-NN is ambiguous, the lack of discrimination about the sense of O_{src} will cause the properties of the wrong sense to be considered for mapping. For example, if knowledge acquisition topic is "rainfall," and one of the near neighbors is the ambiguous concept "shower," then the assertion "showers have plumbing" may become a candidate for being mapped onto "rainfall." The preference of the Map-Props algorithm for questions that have multiple support (that is, a preference for posing questions that were mapped from multiple near neighbors) will normally act against such questions being posed.

Ambiguous words in a property used to select near neighbors. Suppose that a given property of a target object participates in calculating the similarity of the target object to near neighbors. Further suppose that this property contains lexical (or structural) ambiguity. Then, the ambiguous property may give rise to a spurious match to another object. For example, an assertion "showers can be hot" may match the assertion "wasabi can be hot," even if "hot" was used to mean "high in temperature" in the first case and "spicy" in the second case.

If the total number of assertions known about the target object is low, or if several such incorrect matches collude, an irrelevant nearest neighbor will be

returned by Select-NN. If summing evidence in Map-Props does not overcome the presence of the irrelevant object in the near neighbors, or if several similar to each other, but irrelevant near neighbors are present, the system will formulate some (potentially, many) knowledge acquisition questions that are not relevant to the target topic. Alternately, presence of one or more irrelevant objects is likely to bias the system towards posing questions about more general properties, those that are asserted even about the less relevant objects.

Ambiguity in the property of an assertion posed as a question. Suppose that a knowledge acquisition question being posed contains ambiguity (including any of the following: ambiguous words in the property, ambiguous topic of acquisition, or structural ambiguity). Then, prior to being able to answer the question, the contributor has to assign an interpretation to the question that disambiguates this question. This can make the process of answering the question more difficult for the contributor, especially if the contributor is unsure about which interpretation should be answered.

In some cases, the question may be reasonably interpretable in more than one way. Furthermore, the assertion may have different truth values depending on how this assertion is interpreted. For example, consider the two senses of "shower" mentioned above and the assertion "showers are outside." Such cases are particularly difficult because in such cases, the collected knowledge needs to be disambiguated (as discussed in Section 6.4) before the collected truth value can be used.

As the consideration of the possible cases shows, ambiguity in both seed and collected by LEARNER knowledge certainly can affect whether cumulative analogy (in the form used in this work) generates questions that are relevant and easy to answer. In some but not all cases, the "cumulative" nature of the algorithm can prevent ambiguity from having a deleterious effect on the quality of questions being posed.

The next section aims to provide the reader with some feel about the impact of

presence of ambiguity in the collected knowledge base on the usability of the knowledge base for a variety of AI and NLP tasks. I comment on which tasks and approaches are and are not sensitive to ambiguity and speculate on what may differentiate the two kinds of tasks and approaches.

6.3 Tasks and methods sensitive to ambiguity in the knowledge base

In this section, I overview some tasks in artificial intelligence (AI) and natural language processing (NLP) that could use a commonsense knowledge base such as that being collected by LEARNER. For each task considered, my review focuses on the task's need for unambiguous knowledge. The conclusion I draw is that while innovative algorithms and novel approaches may be able to take advantage of ambiguous knowledge, many of the well-known and more straight-forward methods depend on having access to completely unambiguous knowledge. The next section discusses how the amount of ambiguity in the knowledge collected by LEARNER can be reduced.

For a variety of tasks, ambiguity in the knowledge is problematic to approaches which rely on a single assertion to produce their output. For such approaches, a single piece of ambiguous knowledge can lead the system astray.

The classic forward and backward chaining methods of rule-based inference — typically used by expert systems — require unambiguous knowledge. For example, a rule that asserts "if X is a mouse, then X can eat cheese" would lead to incorrect conclusions if the distinction between "mouse" as an input device and a rodent was not made.

Question answering by retrieving relevant assertions from a textual corpus is often approached by searching for appropriate text without performing significant inference (Voorhees, 2000). For this task, the approaches in which a single textual match can produce an answer without regard for context or presence or absence of other (perhaps partial) textual matches. Such simple approaches again rely on a single piece of

evidence to produce their answer, and lexical ambiguity in this piece of evidence can lead the system astray.

On the other hand, some tasks require usage and some approaches take advantage of multiple pieces of evidence in their operation. Such approaches are less vulnerable to lexical ambiguity in the data.

For example, in question answering, accumulating evidence from many retrieved pieces of text reduces the chance that a particular phrasing in a single source will produce a spurious match.

In language processing tasks such as parsing, statistical corpus-based approaches have enjoyed some success, even with very limited amount of knowledge put in apriori. For example, Yuret has demonstrated that structural ambiguity in sentences can be removed on the basis of *lexical attraction* between the particular words in the sentence as observed in the raw examples in the training corpus (Yuret, 1998). This approach exemplifies leveraging a multitude of ambiguous statements to improve correctness of parsing.

In information retrieval, *query expansion* is a technique of improving relevance of retrieved results by expanding the query with additional, relevant terms. It has been shown that expanding the query with related, but not disambiguated (for word sense) terms improves quality of retrieved information (see, for example, Qiu and Frei (1995)).

Finally, by the evidence-summing nature of cumulative analogy, LEARNER itself exemplifies a system that is able to perform its function (knowledge acquisition) by leveraging a multitude of non-disambiguated assertions. The system typically tolerates noise introduced by ambiguity because of the noise suppressing nature or the algorithms Select-NN and Map-props, as discussed in Section 4.2.5.

In summary, across a sampling of tasks, there exist both approaches whose performance strongly requires disambiguated information and approaches that can work gracefully in the presence of ambiguity.

I feel that ability to cope with ambiguity is an important and desirable feature of an algorithm or an approach, due to ubiquitousness of ambiguity and the difficulty of

eliminating it fully. However, resolving ambiguity in the collected knowledge can be an important part of processing knowledge effectively. The following section discusses the methods of doing so.

6.4 Ambiguity of the acquired knowledge can be reduced later

As discussed in the previous section, a number of Artificial Intelligence and Natural Language Processing tasks and approaches benefit from unambiguous knowledge. In this section, I discuss further steps that could be taken to remove ambiguity from the knowledge collected by LEARNER. This discussion is meant to provide a starting point for work that needs to be performed to use knowledge collected by LEARNER in a system that requires unambiguous knowledge.

Broadly, disambiguation of large volumes of ambiguous knowledge at a low cost can be performed by either computer disambiguation programs or, in the spirit of LEARNER and the Open Mind Initiative (Hearst, Hunson and Stork, 1999), by volunteer human contributors. Based on the syntactic restriction that sentences collected by LEARNER do not contain conjunctions, on rarity of complex grammatical sentence structures in the knowledge collected by LEARNER, and my examination of 500 randomly selected assertions, I believe that the most significant kind of ambiguity in the knowledge collected by LEARNER is currently word sense (lexical) ambiguity. Because of this, I focus the remaining discussion mainly to word sense ambiguity. Section 6.4.1 describes a relevant project on collecting word sense information from human contributors and presents some data on the volume and the quality of word sense tagging collected to date by that ongoing project. Section 6.4.2 overviews some relevant approaches and results from the extensive literature on automatic (machine) disambiguation of knowledge.

Section 6.4.1 discusses in greater detail acquiring the information from human contributors, and Section 6.4.2 discusses fully automatic approaches that remove am-

biguity leveraging both disambiguated and ambiguous knowledge.

6.4.1 Acquiring word sense information from human contributors

The most direct approach to disambiguating the collected knowledge may be to turn once more to volunteer contributors on the web. A system designed to perform collection of word sense information from volunteer contributors has been fielded (Chklovski and Mihalcea, 2002). This collaboration between myself and Mihalcea is called Open Mind Word Expert (OMWE), and is available at the time of writing at http://teach-computers.org/word-expert.html.

The project has been fielded in association with the broader Open Mind Initiative (Hearst, Hunson and Stork, 1999). Open Mind Word Expert taps volunteer contributors to assign word senses from WordNet to words that appear in text excerpts. To exemplify OMWE, Figure 6-1 presents a screenshot of OMWE collecting word sense information about the noun "child."

The deployed OMWE system already uses some assertions collected by the Open Mind Commonsense effort (Singh, 2002; Singh, Lin, Mueller, Lim, Perkins and Zhu, 2002), the same source as was used in forming the seed knowledge base for LEARNER. A system such as OMWE could also be directed at disambiguating the knowledge collected by LEARNER.

As further motivation of viability of the approach, I present some statistics about the amount of knowledge collected with OMWE and the reliability of the knowledge collected. In eight months of operation, it has collected a total of 84261 tagging actions. For every item, the system collects redundant tagging — each item is tagged twice by distinct contributors, and agreement between the taggers is tracked, as detailed in (Chklovski and Mihalcea, 2002).

To gauge the quality of the tagging, an experiment to replicate with OMWE some previous tagging of the "interest" corpus has been performed by Mihalcea (R. Mihalcea, personal communication, November 2002). The original "interest" corpus

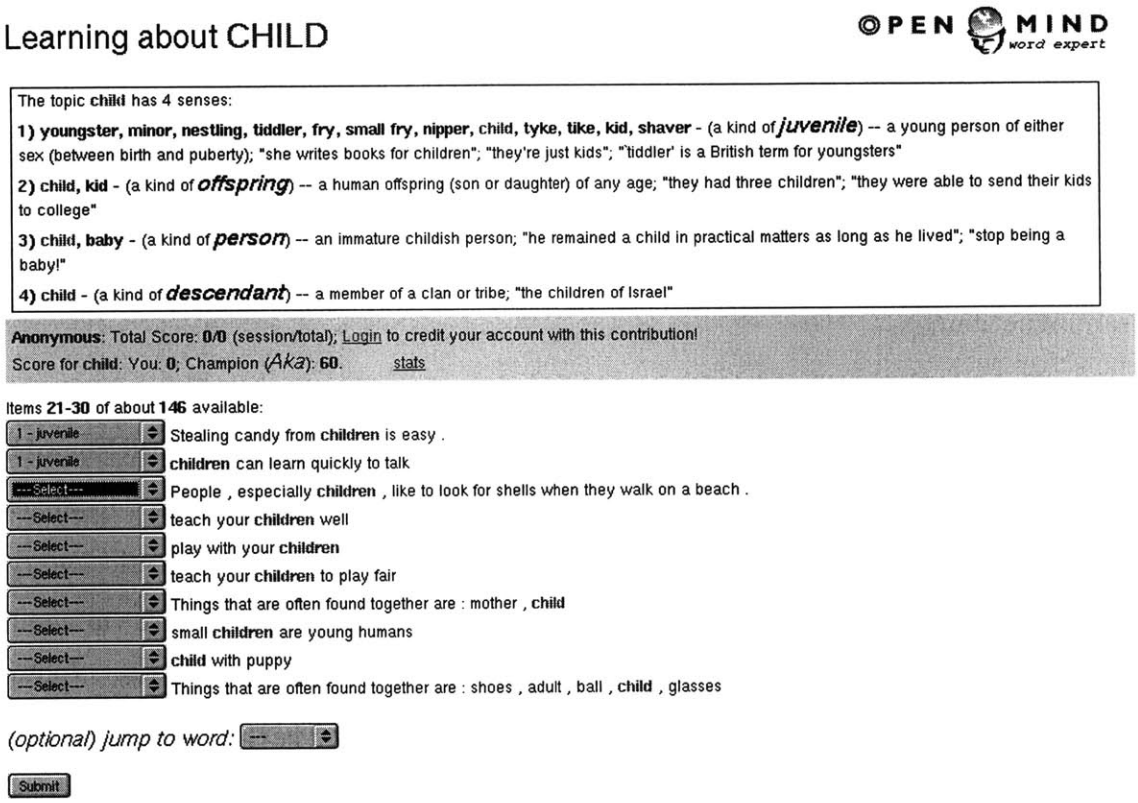


Figure 6-1: Open Mind Word Expert (OMWE): A screenshot of collecting knowledge about "children"

and its tagging are described in (Bruce and Wiebe, 1994). The original tagging was in Longman Dictionary of Contemporary English (LDOCE) senses, and the OMWE tagging was in WordNet senses; the two had to be aligned by producing a map between the WordNet senses and the LDOCE senses.

Out of the 964 items about the noun "interest" for which two OMWE contributors agreed on the tagging, 876 (90.8%) of the tags agreed with the original LDOCE tagging (after the mapping).

This data suggests viability of obtaining some or all of the tagging from human contributors. It does not provide the final answer on what the best way to proceed may be. Further research will be necessary to precisely formulate how such disambiguation should proceed: how many tags per item should be collected, how it should interact with fully automatic tagging, and what inventory of word senses should be used for disambiguation.

One particularly important point to explore is what is the correct "sense inventory," the set of senses into which words are disambiguated. Different dictionaries have distinguish different levels of granularity of a word. Some of the user feedback about the OMWE project has been that WordNet is too fine grained a dictionary, making the task of human annotation too onerous for some volunteers. A viable alternative may be a proposal by Resnik and Yarowsky (Resnik and Yarowsky, 1997), who argue that the desired, non-arbitrary level of granularity is the sense inventory that represent senses that have different lexicalizations present in at least one of a chosen set of languages. For example, if one of the languages considered is French and another English, and if French has two distinct words for what English allows usage of only one word, the two "French" senses should be distinguished in the English word.

Another interesting line of attack may be to formulate knowledge acquisition questions that are less ambiguous in the first place. For example, this could be possible word the word "mouse" by explicitly replacing it in the knowledge acquisition question with "computer mouse," or "live mouse." Although an automatic system may not be capable of finding an unambiguous rephrasing in every case, opportunistic use of this technique may be an effective tool in reducing the amount of ambiguity.

6.4.2 Automatic word sense disambiguation

In this section, I review some results from automatic disambiguation literature and relate them to disambiguating knowledge collected by LEARNER. Traditionally, approaches to automatic disambiguation are divided into two categories: *supervised* or *unsupervised.*

In supervised approaches, a sense tagged corpus of training data is leveraged to disambiguate ambiguous examples. The disambiguation of a word in new text is typically carried out by identifying from the surrounding context a set of features that may be indicative of the sense of the word, and using the exemplars in the training data to find a sense based on these features. The methods used to relate an instance to the training data has ranged from neural networks to naive Bayes classifiers to case-based reasoning. Prior research on word sense disambiguation has also explored a variety of features that may be useful indicators of word sense, and includes surrounding nouns, adjacent (to the ambiguous word) words, parts of speech of the adjacent words, bigrams present in the context of the word and so on. A number of approaches are compared on the word *line* (Mooney, 1996). Additionally, Mihalcea presents a system that selects which features (from a wide set of such features as mentioned above) are useful on a per-word basis (Mihalcea, 2002).

Unsupervised approaches assign senses to a completely untagged corpus. Typically, information about word senses comes from a dictionary entry for this term in a a machine readable dictionary (Lesk, 1986), or from some other starting point provided by a human user — for example, several "indicator" word pairs in which an ambiguous word overwhelmingly assumes a certain sense (Yarowsky, 1995).

The state of the art in word sense disambiguation is reflected by the SENSEVAL competitions.[1] Since 1999, these competitions bring together and evaluate a variety of word sense disambiguation systems on a common set of test data (and, for supervised systems, provide a corpus of training data). The best performing supervised systems, when attempting to disambiguate 100% of the SENSEVAL-2 test data, achieved 64% precision with fine-grained set of senses and 71% precision with a coarser set of

[1] http://www.itri.bton.ac.uk/events/senseval/.

senses. For comparison, trained human lexicographers in creating the training and test corpus, have agreed with their majority vote 85.5% of the time (Kilgarriff, 2002).

Arguably, the most important distinctions to make for using knowledge in reasoning and further NLP processing is to distinguishing between the coarsest senses of polysemous words (word with more than one sense). Yarowsky reports an unsupervised algorithm that achieves more than 96% accuracy on a task involving disambiguation of words with two very distinct senses (for example "plant" in the sense "living plant" or "factory") between pairs of different senses of a word (Yarowsky, 1995). Because of its power and simplicity on this (simpler) binary discrimination task, I briefly describe it here and comment on how it may apply to the data collected by LEARNER.

Yarowsky's algorithm leverages two empirical observations, which I cite from (Yarowsky, 1995):

One sense per collocation: Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship.

One sense per discourse: The sense of a target word is highly consistent within any given document.

To leverage these observations, the algorithm starts from a few (for example, manually) identified seed collocations (e.g. "manufacturing plant" for the "factory" sense, and "plant life" for the living plant sense). The algorithm then uses the two above assumptions to bootstrap from the identified instances to generate new instances, identifying new collocations in documents where known collocations are present. The task of disambiguating knowledge collected by LEARNER differs in the following way: the collected knowledge is not organized in larger documents. Rather, it exists in sentences that are similar to one another. I speculate that similar sentences collected by LEARNER could be used to play a role similar to the role of larger documents in Yarowsky's algorithm. Clearly, additional investigation would be necessary to conclusively establish applicability of such methods.

As a closing remark, I point out that delaying disambiguation of knowledge may reduce the total effort expended. This is because *many imprecise assertions can, together, become more precise*, as exemplified, for example by Yarowsky's bootstrapping algorithm (Yarowsky, 1995).

Additionally, manual disambiguation may be combined with automatic disambiguation in a manner that makes best use of human effort for tagging carried out partially manually and partially automatically, both in terms of the amount of manual effort and the overall precision of the resultant tagging. To derive better improvement of a supervised algorithm from human disambiguation effort, a kind of *active learning* can be employed (Dagan and Engelson, 1995). Specifically, the machine learning algorithm could identify the data that would be most useful to learning and request human tagging of such data, a method that has been applied in Open Mind Word Expert (Chklovski and Mihalcea, 2002). If the automatic disambiguation algorithm is capable of not providing an answer on instances that are most likely to be tagged incorrectly by the algorithm, human effort could be directed to those, raising overall precision. In light of the extensive prior work in the field of automatic word sense disambiguation, combined with the possibility of attaining large amounts of human tagging from volunteer contributors, I believe the prospects for disambiguating the collected knowledge are good, even though additional work is required to reduce these observations to practice in a single disambiguation approach.

Chapter 7

The Correlated Universe, or Why Reasoning by Analogy Works

A lot of the knowledge acquisition power in this thesis comes from posing questions by analogy. But why should analogy allow for useful mappings of properties?

Fundamentally, the underlying assumption behind reasoning by analogy is that *properties of objects in the world (as reported by humans) are correlated.* That is, for any given object O there will likely exist some objects $O_{s_1} \ldots O_{s_i}$ that share more properties with can be expected by chance. For example, both a "dog" and a "cat" have the following asserted about them: "has a tail," "eats meat," "is a pet." Under LEARNER's criteria for similarity, which will be explained below, "dog" and "cat" share 42 properties in the knowledge base, whereas by chance they can only be expected to share 6. Note that human cognition introduces a bias as to what set of properties is used to compare objects. The influence if this factor is discussed further in Section 7.5.

In this chapter, I investigate similarity, analogy, and amount of correlation quantitatively. This investigation allows us to better gauge the power of the class of algorithms based on cumulative analogy. Three methods of analysis are introduced, each characterizing the knowledge base and the effectiveness of cumulative analogy from a different angle. This investigation:

- quantifies the amount of similarity in the knowledge base;
- derives evidence that reasoning by analogy is a well-motivated knowledge acquisition approach;
- derives a lower bound on how much analogy can accomplish;
- points out limitations of reasoning by analogy.

To establish these results, three methods will be used:

Average similarity histogram This analysis calculates with how many objects, on average, a given object shares one property, with how many it shares two, and so on.

Reach of analogy This analysis looks at how far analogy can get us. That is, if a single property in the knowledge base is held out, in what percentage of cases can it be established by analogy, making specific choices about values of parameters that a simplified analogy algorithm (without the elaborate weighting of properties used in Select-NN) would take as inputs.

Nearest-neighbor distance This analysis studies nearest-neighbor distance. It measures, for each object, how similar the most similar object is.

All three analyses shed light on the amount of correlation in the knowledge base and on the expected applicability of reasoning by analogy. The first and second analyses only look at assertions of the "is true" variety. The third analysis also accounts for the "is false" assertions.

Additionally, the analogy-based approach can be used not only to pose knowledge acquisition questions, but also as a reasoning method — guesses about the truth or falsehood of an assertion can be made using the same analogical reasoning. The third analysis speaks to the correctness of the predictions made by reasoning by analogy.

7.1 Overview of the knowledge base

Recall that Chapter 3 described the way knowledge is represented in LEARNER internally. Before delving into the subject of correlation deeper, I provide some statistics about the knowledge base being analyzed (the "seed knowledge base," as is explained below).

In analyzing applicability of reasoning by analogy to the seed knowledge base, some simplifying assumptions are made.

The chief simplifying assumption made in my analysis is that each sentence is considered here only as a subj-assertion (i.e. an assertion about the sentence's syntactic subject having a property). For example, "cats have tails" is, for the sake of my analysis, treated as the object "`cat`" having the property "`have tail.`" Doing so allows us to avoid some thorny issues about double-counting assertions derived from the same sentence.[1]

The other difference between the analysis and the algorithm is that the analysis is performed only on the seed knowledge base (the knowledge gathered without using analogical reasoning by another project, and that served as the starting point for LEARNER).

The seed knowledge base was created primarily by asking contributors to state something, or to fill in the blank by an effort predating LEARNER and without my participation (Singh, 2002). Contributors were never asked to specify a truth value. Only later linguistic processing of assertions such as "birds can fly" and "a worm does not have legs" interpreted them as $A(\textit{bird}, \textit{can fly})$ and $\neg A(\textit{worm}, \textit{have leg})$.Recall that I denote that the object O_i has the property P_j by writing $A(O_i, P_j)$, with A signifying "assertion."

Because of the approach to knowledge base collection, the overwhelming majority of the assertions (96.0%) are "is true" assertions. Note that when several statements map to the same assertion (for example, "a cat has a tail" and "cats have tails"), only one of the statements was automatically selected for inclusion in the analysis.

[1]See Chapter 3 (Representation) for a description of different kinds of assertions and how they are computed.

			Columns (Properties)			
			with ≥ 10 entries	with ≥ 2 entries	with one entry	Total
			291	4905	28070	**32975**
			1%	15%	85%	**100%**
Rows (Objects)	with ≥ 10 entries	723 6%	*2108* *4%*	*9282* *20%*	*17119* *36%*	*26401* *56%*
	with ≥ 2 entries	4277 35%	*4091* *9%*	*15252* *32%*	*23846* *51%*	*39098* *83%*
	with one entry	8049 65%	*1499* *3%*	*3825* *8%*	*4224* *9%*	*8049* *17%*
	Total	**12326** **100%**	*5590* *13%*	*19077* *40%*	*28070* *60%*	**47147** **100%**

#Entries(Assertions)

Table 7.1: Summary of the seed knowledge base. Total numbers of objects (rows), properties (columns) and entries (assertions) are in bold. Other counts are for rows, columns and entries when only indicated subsets of rows of columns are considered. For example, there are 723 objects with at least 10 properties, and 26,401 assertions about these objects. For clarity, the "is false" entries are not included in these results.

The essential statistics, including the number of distinct objects, properties, "is true" assertions and more are presented in Table 7.1. The analysis focuses on "is true" assertions, returning to the "is false" assertions in Section 7.4.

There are several things to note about Table 7.1. One observation is that for 65% of the objects present, only one property is known about each such object. The assertions about objects with only one property constitute 17% of all of the assertions. Similarly, 60% of the assertions are asserted about only one object (and thus cannot, for example, be expected to be mapped from more than one near neighbor).

These numbers suggest that a fairly large part of the seed knowledge base is sparsely populated with assertions. More specifically, the seed KB has a densely populated "core," tapering off to large sparsely populated regions. At the same time, cumulative analogy is a method that requires a significant amount of priming, and, as is, it will meaningfully apply only on a fraction of the objects in the knowledge base. For instance, the result of applying cumulative analogy to objects about which only one property is known is not likely to yield good results. At the same time, the

applicability of cumulative analogy to any given object about which little is known should be improved by specifying more about such an object. Specifying approximately ten additional properties about an object about which little is already known is, in my experience, quite easy and is supported by the LEARNER interface.

To further characterize the distribution of knowledge in the seed knowledge base across objects, Figure 7-1 presents a log-log plot of the numbers of objects with one, two, and so on properties known about each object. The values were fitted by an expression of the form $f(x) = Cx^p$. The values of the two parameters, C ($C = 6789$) and p ($p = -1.9483$) were chosen by fitting the data for $1 \leq N \leq 50$ to minimize the sum square difference of *logarithms* of real and fitted values. The fact that $p \approx -2$ suggests that the distribution fits Lotka's law (Lotka, 1926). This approximation for the number of objects with n properties, related to Lotka's law, is also used in Appendix D.

Lotka has found that the number of authors making n contributions in chemistry is approximately $1/n^2$ of the number of authors making 1 contribution. This observation has consequently been found to apply in a number of other fields — for example, the number of incoming and outgoing links of web documents. Lotka's law is also closely related to Zipf's law (Ye-Sho Chen, 1986).

Zipf's law[2] (Zipf, 1949) is the observation that for a variety of phenomena from frequencies of words in a text to populations of cities, the frequency of an event P can be expressed as a function of its rank i according to the power-law function $P_i \approx 1/i^a$ with the exponent a close to 1. Note that while Lotka's law addresses the number of authors with one, two, and so on publications (or objects with one, two and so on properties), Zipf's law addresses the number of properties of an object with the most properties, the number of properties of the second-ranking (by number of properties) object, and so on.

[2]See http://linkage.rockefeller.edu/wli/zipf/ for an extensive list of resources concerning Zipf's law.

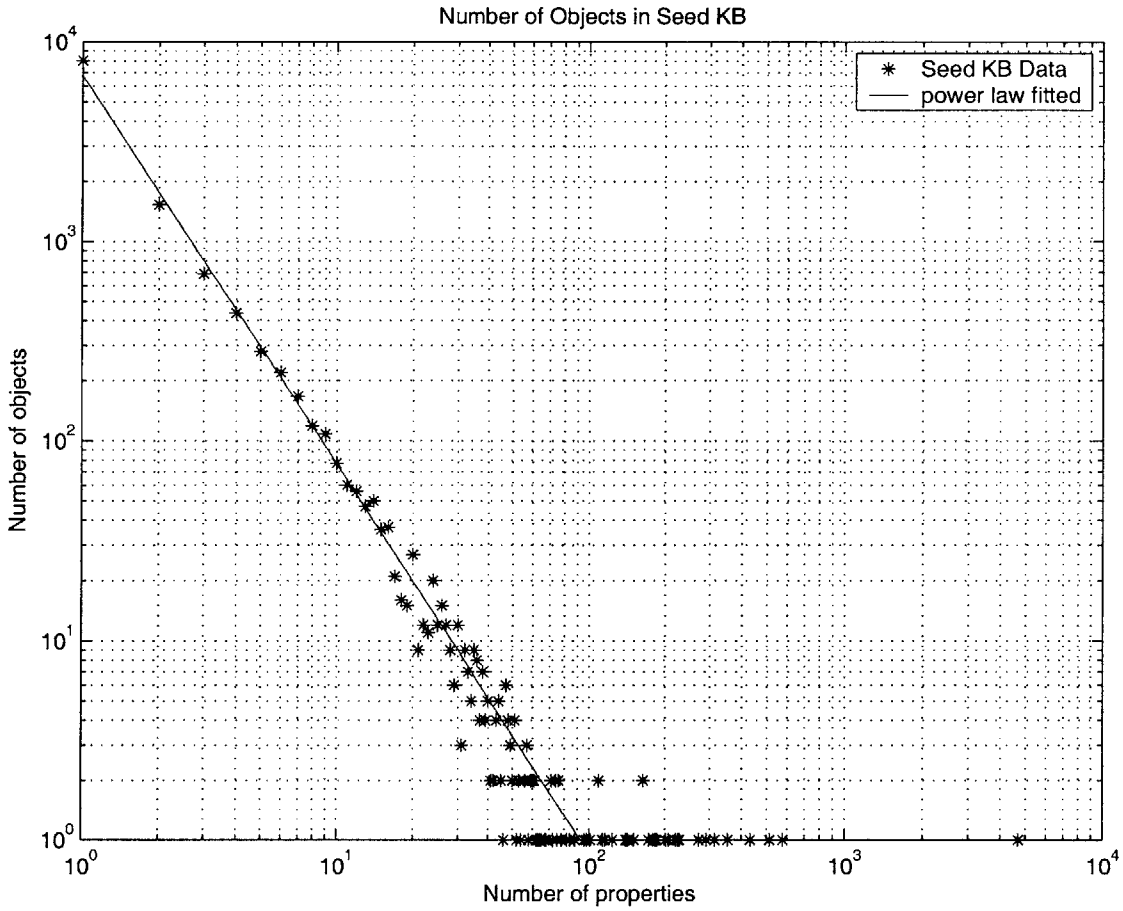


Figure 7-1: Numbers of objects with N properties in the seed knowledge base on a log-log scale. The solid line is an expression $f(x) = Cx^p$. C ($C = 6789$) and p ($p = -1.9483$) were chosen by fitting the data for $1 \leq N \leq 50$ to minimize the sum square difference of *logarithms* of real and fitted values. The fact that $p \approx -2$ suggests that the data fits Lotka's law (Lotka, 1926), which is closely related to Zipf's law (Zipf, 1949).

7.2 Amount of similarity

How much correlation is present in the seed knowledge base? To the extent that the seed knowledge base is a collection of assertions that people hold to be important about objects in the world, the amount of correlation in the knowledge base reflects how correlated a world is being described in it. One can imagine both a very correlated world in which knowing just a few properties of an object enables predicting very much about it. Conversely, one can imagine a chaotic world where very little can be derived by analogy. Where on this spectrum does the seed knowledge base lie?

To make progress on the issue of correlatedness, the question can be reformulated more concretely:

> If an object is selected at random, with how many other objects would it share one, two, three, ..., twenty properties?

Figure 7-2 presents, for N from 1 to 20, a histogram of the number of objects with which given object, on average, will share exactly N properties. As a baseline, a histogram is provided for an artificially generated "uncorrelated world" with the same number of objects, properties and with the properties obeying the same frequency distribution as in the real knowledge base.

Note that Appendix D derives a closed-form approximation for the "uncorrelated" case. However, deriving the approximation has required making simplifying assumptions about the number of objects with one, two and so on properties. Here, I instead present the results for the decorrelated baseline case derived by simulation, as follows: for each object O_i, the properties it contains were "decorrelated." That is, if a property P_j held for fifty other objects, fifty objects from all the known objects were selected at random under a uniform probability distribution and counted as having this property. Then, the same calculation as for the original case (how many objects it shares N properties with, for different values of N) was performed. In both the decorrelated and the original cases, a histogram was computed by averaging across all objects.

The result is presented in Figure 7-2 (the Y-axis is on a log scale). The histogram

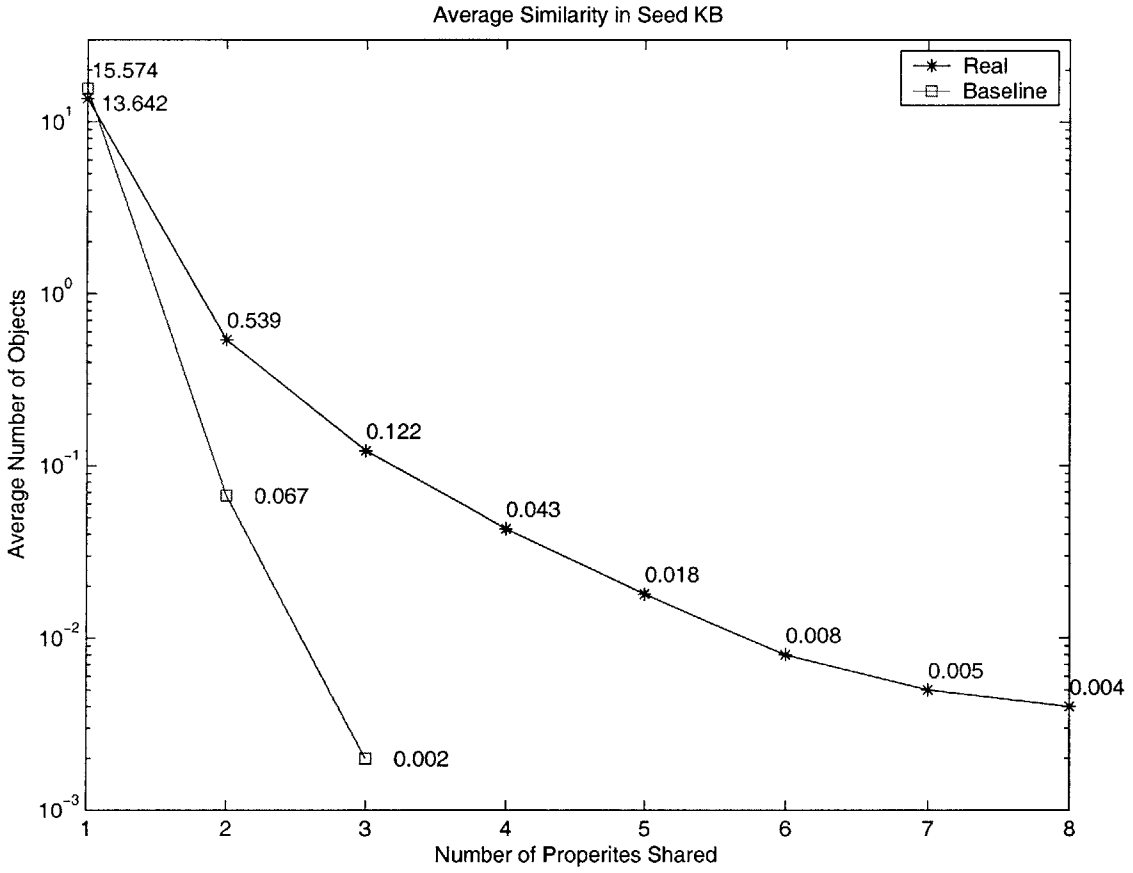


Figure 7-2: Average correlation histogram for the seed knowledge base. Expected number of objects sharing one, two, and so on properties with a given object the amount in the real and synthetic, (uncorrelated) case with the same frequency distribution is shown. The histogram plots, for different values of N along the X-axis, the number of objects with which a given object is expected to share N properties (on the log-scale Y-axis). The data represents the average for 12,326 objects in the knowledge base.

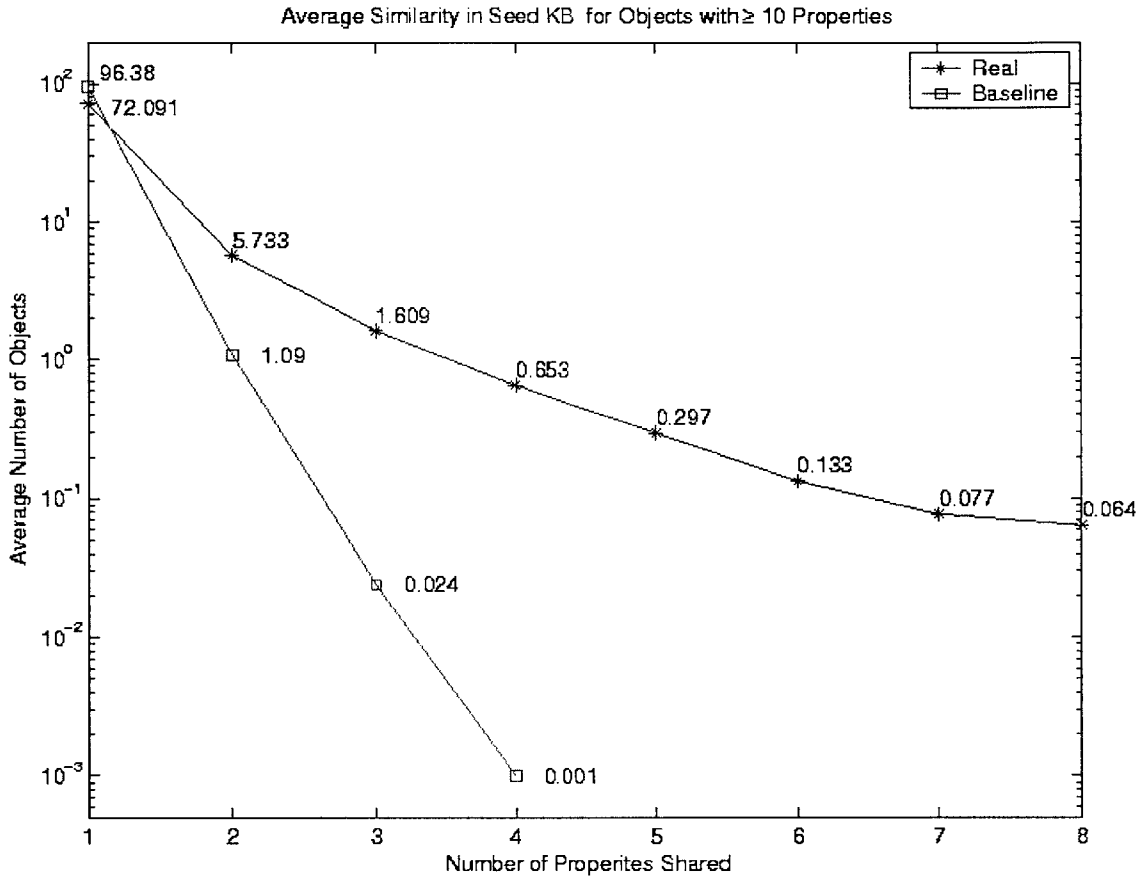


Figure 7-3: Similar to Fig. 7-2, calculated only over objects with 10 or more properties. Y-axis is, again, logarithmic. For each of such 723 objects, similarity with all 12,326 objects was measured. Notice that the number of similar objects decays more slowly in the real case compared to the "no correlation" case.

shows that on average, each object has more similar objects than can be expected by chance. On average, sharing two properties is eight and three properties is fifty times more likely than chance. Because the properties are correlated with each other, the average number of objects with which exactly one property is shared is actually lower than could be expected by chance. On average there are approximately 13.6 such objects rather than the expected by chance 15.6 (note that the lines in Figure 7-2 cross over between one and two, almost at one). Also note that, on average, there is less than one object sharing more than two properties with a given object. This phenomenon is due in part to many objects in the seed KB having only one property. The amount of correlation that exists for objects about which more is known is addressed by the following analysis.

As mentioned above, cumulative analogy is a method that needs sufficient priming to apply. To better illustrate the average similarity of objects with many properties already known about them, objects with ten or more properties are analyzed.[3] Figure 7-3 presents results of computing the average similarity histogram ranging over objects with ten or more properties.

In this case, the amount of similarity between objects is even more pronounced. For each object of interest one can expect, on average, to have 1.6 objects that share three properties, on average at least one object that shares four or more properties with a given object. These numbers suggest that, for objects about which ten or more properties are already known, cumulative analogy is likely to find at least some similar objects from which to map properties meaningfully.

7.3 Reach of analogy

In this section, I quantify what percentage of knowledge can be established by generalization by analogy.

Reasoning by analogy can be viewed as mapping properties from similar objects

[3]The specific threshold of ten used to have special significance in earlier versions of the system, but currently does not.

onto a given object. The absolutely simplest case of analogy is two objects sharing a property, giving us grounds (however small) to map other properties of one object onto the other.

This observation inspires a definition. Let's call property P of object O_1 *directly acquirable by analogy on K properties* if (i) there is an object O_2 that also has property P, and (ii) O_1 and O_2 share at least K other properties.

Note that the set of assertions *potentially* acquirable by analogy is the superset of the directly acquirable assertions. This is because acquiring assertions about an object O_1 by analogy with some O_2 can cause O_1 to become sufficiently similar to some other O_3 to permit additional analogies. I obtain a lower bound on potential acquirability by calculating direct acquirability.

The question addressed is: what percentage of assertions are directly acquirable by analogy? That is, if one assertion is held out at random, for what percentage of cases is it directly acquirable (i.e., for what percentage of assertions will the algorithm inquire about their truth)?

Some preliminary observations will help us frame this question correctly. In a growing knowledge base, some properties will be known about only one object. Note that these properties could never be acquired if held out, because holding them out removes any trace of their existence. Some examples of properties that hold for only one object are as follows:

- "a caterpillar will turn into a butterfly"
- "heart is responsible for pumping blood"
- "typewriters have been mostly replaced by computers"
- "Frascati is an Italian wine"
- "San Francisco is west of Texas"

The above examples suggest that some singleton properties (such as "`will turn into a butterfly`") would not be acquirable by direct analogy from other assertions by being mapped verbatim from other objects (because no other objects turn

into butterflies), while other properties such as "`is an Italian wine`" are probably known about only one object in the seed knowledge base due to incompleteness in its coverage.

In the seed knowledge base, there are 28,070 (60%) such assertions. *All calculations in this section are restricted to the remaining 40% of assertions* — those that would be directly analogizable in a knowledge base describing a fully correlated world.

Furthermore, analogy cannot be expected to apply to objects about which only one property is known. Restricting attention to objects with two or more properties, and asking what is the direct acquirability by analogy on one property, it turns out that 58.8% of 15252 such assertions are directly acquirable. Using the notation of $Reach(N, M)$ to denote the percentage of directly acquirable assertions about objects with at least N properties by analogy on M or more properties, the above can be written as:

$$Reach(2, 1) = 58.8\% \tag{7.1}$$

Let us investigate some variants of reachability. Restricting attention only to objects with at least 10 (possibly unique) properties, and asking questions by analogy based on two properties being shared (direct acquirability by analogy on two properties) results in a universe of 9282 assertions being considered for direct acquirability. It turns out that 5239 (56.4%) of these assertions are directly acquirable. To put it another way,

$$Reach(10, 2) = 56.4\%. \tag{7.2}$$

Note that objects with any number of properties are permitted to serve as *sources* of analogy in this analysis.

Considering only objects with at least 20 properties,

$$Reach(20, 2) = 64.7\% \tag{7.3}$$

of the total of 6958 such assertions are directly acquirable.

Even requiring the source and target of analogy to share at least 4 properties, it

turns out that

$$Reach(20, 4) = 45.9\% \tag{7.4}$$

of these assertions are reachable.

Adopting datamining terminology (Agrawal, Imielinski and Swami, 1993), so far the data has concerned assertions reachable with support of one (that is, the model was that having just one object of sufficient similarity was assumed to warrant posing a question). An alternative is to require that there be at least k objects sharing M properties to warrant posing a question. Let us use the notation $MultiReach(N, M, k)$ to denote the percentage of existing assertions about objects with at least N properties which are reachable by analogy on M properties from at least k objects. Given this definition, *Reach* can be viewed as a special case of *MultiReach*, with $MultiReach(N, M, 1) = Reach(N, M)$.

Considering a property of an object O_1 reachable only if there are at least two objects O_2 and O_3, each sharing at least one other property with O_1, and looking at objects with at least 10 properties, it turns out that 38.3% of properties are reachable under this definition. To put it another way,

$$MultiReach(10, 1, 2) = 38.3\% \tag{7.5}$$

Overall, reachability between 38.3% and 64.7% has been observed for different strictnesses and scopes of reachability. Note that the analysis was restricted to the objects with sufficient number of properties, with the threshold for the minimum number of properties known varying between 2 and 20. It seems that the degree of similarity between objects (presented in the histograms of Section 7.2) does indeed bear out in the analysis of reach of analogy, allowing the analogical approach to achieve the reported degree of reach in a real knowledge base. The results of the above three methods of analysis suggest that acquisition by analogy from several nearest neighbors is likely to be successful given a fairly large and correlated knowledge base as a "seed knowledge base" (such as the seed KB analyzed here). Chapter 8 reports on the results of an empirical verification of this claim.

It must be noted that this analysis excluded a relatively large set of properties that hold about only one object. The inability to acquire by analogy properties mentioned zero times is an important limitation of the approach.

One extension that I believe can be a powerful next step beyond direct analogy and can satisfactorily address this issue is discussed in Chapter 9.2 (Future Work). The proposed extension would formulate knowledge acquisition questions using the current algorithm, but run it over a more abstract representation of the assertions present. For example, recognizing that "sugar is sweet" as asserting "has a (specific value of) taste" and "Tabasco sauce is spicy" as stating that "has a (specific value of) taste" could lead the system to infer by the current mechanism that, for example "`salt`" or "`pepper`" also have specific values of taste. The system could then ask what that value is (i.e. ask how salt or pepper taste). This would allow LEARNER to tap related, not only identical properties in posing its questions.

7.4 Similarity of most similar

Another way to characterize the amount of correlation in the world (as it is reflected in the knowledge base) is by looking, for each object, at its similarity to the object most similar to it.

The analysis of similarity of the most similar allows us to study more rigorously the expected performance of a simple analogy algorithm based on the single nearest neighbor. This complements the previous section's more theoretical study of analogy by studying its direct reach. Also, this analysis will allow us to address the "is false" assertions.

First, a bit of notation: objects O_1, O_2 are said to share a property P if $A(O_1, P) \wedge A(O_2, P)$. Objects are said to mismatch on a property P if $(\neg A(O_1, P) \wedge A(O_2, P)) \vee (A(O_1, P) \wedge \neg A(O_2, P))$.

The number of properties shared with nearest neighbor in the seed knowledge base is presented in Figure 7-4. This figure plots average results for objects with 2,3,..., 25 properties along the X axis. The Y axis represents the percentage of properties shared

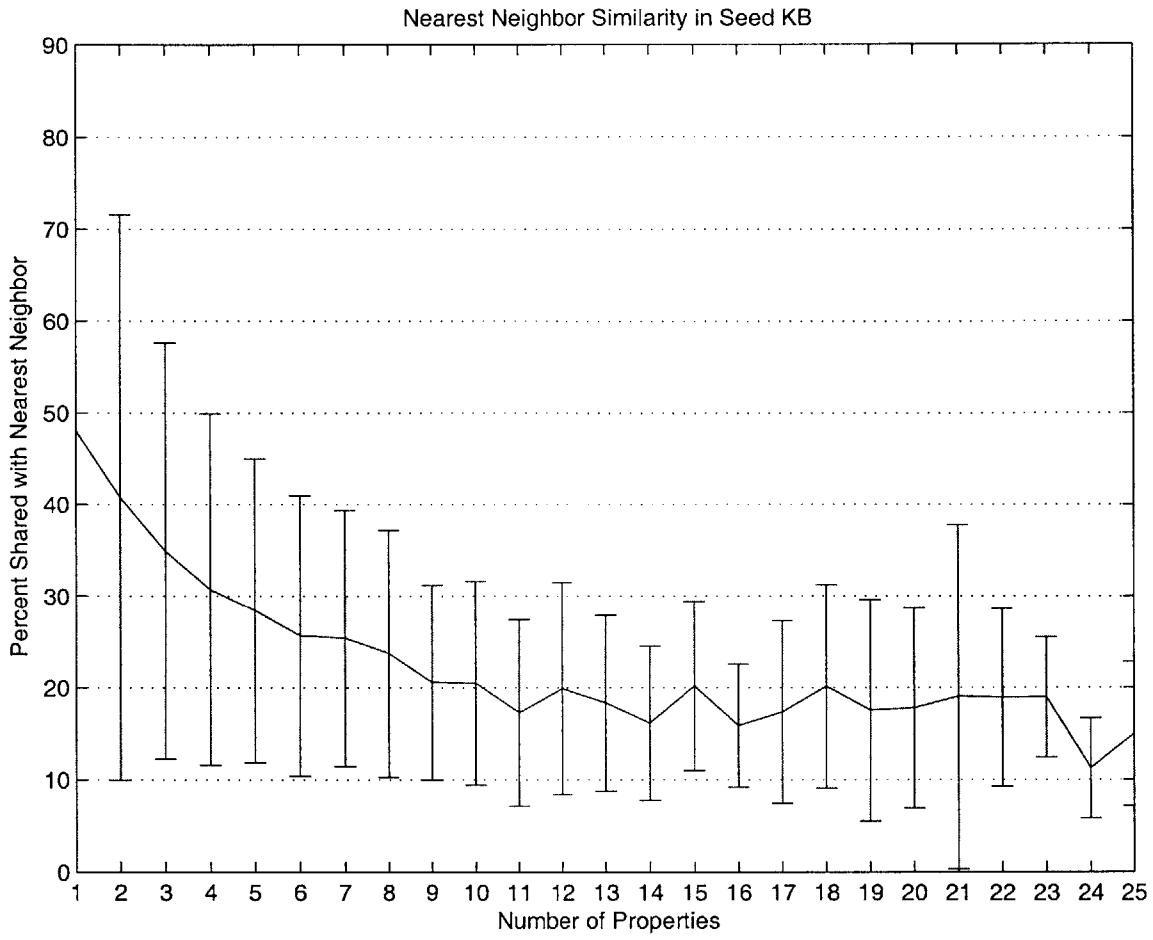


Figure 7-4: Nearest neighbor analysis. For each object, the percentage of "is true" properties shared with its nearest neighbor in the seed knowledge base has been measured. The X-axis represents the number of properties of an object (results for objects with same number of properties have been averaged), so the values for objects with different numbers of properties can be observed.

with the nearest neighbor. Error bars of one standard deviation in either direction are also plotted. The nearest neighbor for a given object was computed by selecting the object that shares the most "is true" properties, breaking ties deterministically.

The histogram shows that for objects with more than five properties, an object can be expected to share about 20% of its properties with its nearest neighbor. Note that unique properties are included in the denominator in this analysis. The figure also shows that 68% of such objects (those within one standard deviation of the mean) share between 10% and 30% of their properties with their nearest neighbor. The general pattern is violated by objects with twenty-four properties, which may be

attributable to scarcity of the data.

Recall that 60% of the assertions in the seed knowledge base assert a property that holds for only one object. These assertions clearly will not be shared with the nearest neighbor. When these are excluded from consideration, it turns out that there are 19,077 assertions that assert properties that hold for more than one object (see Table 7.1), and 11,190 (58.6%) of these assertions are shared with the nearest neighbor. Once again, this points to presence of strong correlations in the data, and motivates application of nearest neighbor based techniques to such knowledge acquisition tags.

7.4.1 Mismatches

In this section, I examine the number of mismatches of truth values of identical properties that an object has with its nearest neighbor. For the purposes of this analysis, nearest neighbors are computed by attending solely to "is true" assertions.

As mentioned above, objects share a total of 11,190 "is true" properties with their nearest neighbors (of a total of 47,147 assertions in the seed knowledge base). They also agree with the nearest neighbor on an "is false" property for 170 properties (of a total of 2,482 "is false" properties in the seed knowledge base). Finally, they mismatch on a property in 593 cases, 257 of them being the nearest neighbor having an "is false" and the object itself having an "is true" property.

This information allows us to answer an interesting question. Namely, if the nearest neighbor is used to predict properties of an object, how often would the prediction be correct? (Disregarding the effect that holding out an "is true" property could change the nearest neighbor). This analysis allows some insight into expected success of using nearest neighbors not only to posing questions, but to actually use them in analogical reasoning to predict what is and is not true.

If a single nearest neighbor is determined using all properties of the object, and then truth values of its properties are used in lieu of the known truth values of properties asserted about the target object, the truth value of an "is true" property would be predicted correctly in 11,689 cases, and incorrectly in 257 (correct 97.8%

of the time). The truth value "is false" would be correctly predicted in 170 cases and incorrectly in 336 cases (correct 33.5% of the time). Overall, we'll be correct in 95.2% of the cases. Note that these numbers, including the correctness overall, are affected by the dominance of "is true" assertions in the knowledge base. The baseline strategy of always guessing "is true" for every assertion would correctly guess 100% of "is true" and 0% of "is false" assertions, being correct in 95.9% of the cases. In effect, predicting from the single nearest neighbor has traded off correctly predicting 170 truth values of "is false" assertions for incorrectly predicting 257 "is true" assertions. These numbers point to the bias that exists (and perhaps the limitation of the "single nearest neighbor" strategy examined) in the case of a knowledge base unbalanced in terms of proportion of "is true" and "is false" assertions. It would be interesting to see what the picture looks like in a more balanced knowledge base which has not been biased by using analogy to extend it.

In practice, a hypothesis can be formed by combining evidence from several near neighbors (as the implemented KA algorithm does), which should have both higher coverage and more accurate predictions.

Discovering exceptions in truth values (such as "penguins can fly" is false) is part of an accurate description of the world and hence is part of the desired set of the assertions to acquire. I expect that acquisition using nearest neighbors to uncover many "near miss" (Winston, 1972) mismatches in truth values that were not elicited by the methodology used to construct the seed knowledge base, which contains predominantly "is true" assertions.

7.5 On the origin of similarity

In the previous sections, I have addressed the amount of similarity. In this section, I focus on the epistemological sources of similarity.

What are the origins of similar assertions being made about similar objects? It seems that at least two sources of similarity can be identified.

Synonymy Similar things will be stated about "`car`" and "`automobile.`" The origin

of this correlation is lexical. (However, note that the word "car" has usages which "automobile" does not — for example, the seed knowledge base contains the assertion "trains pull cars," referring to the "railroad car" sense of car).

Taxonomic proximity This is probably the most familiar kind of similarity. A "cat" and a "dog" are both "pets" and "animals", "TV" and "radio" are both "consumer electronics," and "apple" and "pear" are both "fruit." Some objects in the world (animals, plants) are similar for evolutionary reasons, and other, man-made objects are similar because they were manufactured to serve similar functions in our lives.

In addition to similarity between objects (two objects sharing many properties), there is also a dual situation of a pair of properties being correlated across many objects. The correlation between properties seems to be classifiable into three classes. Two of these classes parallel the causes for similarity between objects: they are synonymy and taxonomic proximity between terms used in properties.

Similarity of properties due to synonymy is exemplified by the following: the verb "hold" can mean the same as "contain," which gives rise to pairs of correlated properties such as "can hold water" and "sometimes contain water", which are both true of "cups," "bottles," and "bathtubs," and are both false of "books," "keys," and "knives." Taxonomic similarity of properties is exemplified by the following: "rivers" and "seas" are both kinds of a large body of water, giving rise to correlated assertions "(fish) live in the river/sea", "(yachts) sail on rivers/seas", "(a person) can go swimming in a river/sea".

Unlike objects, however, a property asserts something about an object. Because of this greater complexity, there is an additional feature of properties. They can be correlated because one semantically implies the other. For example, "have wings" and "can fly" are correlated. Sometimes properties are correlated because they have a common cause, as in this example: for comfort, most enclosures made for humans typically require both a way to enter and exit the enclosure, and a way to observe the outside. Hence, enclosures that "have doors" (such as houses, cars, airplanes),

usually "`have windows`" as well.

Based on the observation that there is property-based correlation in addition to object-based correlation, it should be possible to pose questions by similarity of *properties* rather than *objects*. For simplicity, similarity of two properties can be measured by the number of objects that they are both asserted about. It turns out that while there are more properties than objects, direct acquirability from correlation of properties can be as high as direct acquirability from correlation of objects.

For example, restricting to objects with 10 or more properties and properties that hold for 10 or more objects, direct acquirability by analogy on two properties (based on object similarity) is 68.7% (1448 of 2108 assertions). Under the same restrictions, the direct acquirability by two objects (based on similarity of *properties*) is 69.4% (1462 of 2108 assertions). Considering the similarity by four rather than by two on the same set of assertions results in direct acquirability of 37.3% and 34.0% for the "by objects" and "by properties" cases, respectively.

It is possible that combining these two methods would yield better question quality than each method alone. Even more importantly, presence of the observed degree of correlation between properties indicates that it may be possible to eliminate posing "redundant" (highly correlated) questions about an object in the knowledge acquisition interface.

Overall, it seems that similarity stemming from all of the described causes is present in the knowledge base. The knowledge base exhibits a degree of similarity that justifies approaching knowledge acquisition via nearest neighbor methods. Similarity stems from semantic relationships between pairs of objects and between pairs of properties, and exploiting both may yield more effective knowledge acquisition methods.

When two categories of objects share many properties, such categories are similar. However, to the extent that these categories do not describe the same objects, there will also necessarily be differences between the categories. The need to discern the differences (to reason correctly about a given category or situation) is what gives rise to the categories in human descriptions of the world in the first place. As has

been explored by Winston (1978), when a category is used as a source in a metaphor (an analogy), the features being transfered are likely to be those that differentiate the source category from the similar to it categories. For example, when the term "`fox`" is used in the assertion "Robbie is like a fox," what is meant is probably that "Robbie" has the features that distinguish a "fox" from other animals. That is, given some conventional knowledge about foxes, Robbie may be particularly clever or cunning.

Because differences between categories are innate to the notion of categorization, it can be expected that every pair of categories should have some mismatching features and that most categories should have characteristic features.

Having discussed the origins of similarity and presence of differences, I examine the issue of similarity judgments being dependent on the set of features chosen to describe the objects being compared. It seems that the choice of features, even when it expresses equivalent information, can affect which objects are considered similar and which are not.

Consider the following:

> Suppose we use the two features `blind-in-left-eye` and `blind-in-right-eye`. Then the four possible objects are {0, 0}, {0, 1}, {1, 0}, and {1, 1}, with their obvious interpretations. Suppose Alan is blind in his left eye only {1, 0}, Bob blind in his right eye only {0, 1}, and Charlie blind in both eyes {1, 1}. Then Alan and Bob are equally dissimilar to Charlie (according to [LEARNER's similarity] measure). But suppose instead we employ an equivalent representation: `blind-in-left-eye` and `same-in-both-eyes`. In this second representation the four objects above are represented: {0, 1}, {0, 0}, {1, 1} and {1, 0}. Now Alan {1, 0} and Bob {0, 0} are not equally dissimilar to Charlie {1, 1} (personal communication, D. Stork, 2002). See also Duda, Hart and Stork (2000, pp. 458-461).

This issue is related to the well known "theorem of the ugly duckling" (Watanabe, 1969) which states that, given a set of objects and a set of features that allows any

two objects being compared to be distinguished, in the absolute absence of bias any two objects share the same number of *predicates* defined over the set of features. The theorem derives its somewhat fanciful name from the counterintuitive illustrative example that, in the absence of bias, given two ducks and one "ugly duckling," the two ducks are equally dissimilar from each other as each one is from the "ugly duckling."

The approach adopted in this work, however, does introduce a strong bias. Similarity is measured by computing (weighted) sums of the number of matching features and subtracting an (also weighted) number of mismatching features. By counting the number of matching features rather than the number of all predicates formed from the features, comparisons are made according to the simple constructs (features) that the human contributors have formulated.

The issue of using slightly different features resulting in different similarity judgments has a more direct relevance to LEARNER's similarity judgments. Indeed, it seems that in some cases the exact set of features can make some pairs of objects more and some less similar. However, the investigation in Sections 7.2 and 7.3 has provided a *prima facie* argument that similarity between the collected sets of features can be used to successfully predict presence of other features.

Overall, in this work, I take a descriptive approach to common sense, seeking to collect salient elicitable knowledge about concepts, rather than, for example, treat concepts as predicates which need to be minimally defined in terms of other predicates to be distinguishable from other concepts. By taking the more descriptive stance in this work, I collect descriptive knowledge thought to be the correct level of description in the *naive semantics* tradition. Naive semantics is further contrasted with formal logic and some "semantic primitives" approaches in Dahlgren et al. (1989).

Chapter 8

Results

In this chapter, I (i) present evidence that LEARNER's knowledge acquisition power indeed stems from reasoning by analogy from similar objects, and (ii) present the results of running the LEARNER on a public web site

http://teach-computers.org/learner.html

over the course of two months.

Specifically, the following data is presented:

- How the quality of the questions compares with the quality of questions generated by an ablated version of the algorithm that poses questions without using any notion of similarity,
- The amount and kinds of knowledge collected, (percentage of taxonomic statements, part-of statements, etc.), and how it compares with data available for other wide coverage knowledge bases,
- Overall statistics of contributor behavior (average amount contributed in one visit, number of repeat visits from the same computer, and so on).
- The contributor feedback that was gathered as the system was run,
- My own impressions (with discussion) of the system's current limitations.

In my approach to evaluating knowledge acquisition, some prior work has served as an inspiration. A relatively recent discussion of methodology for evaluating knowledge acquisition by Tallis et al. can be found in (Tallis, Kim and Gil, 1999). Prior investigations by Cohen et al. and by Gaines (on different knowledge bases) have quantitatively studied the usefulness of prior knowledge for knowledge acquisition (see (Cohen, Chaudhri, Pease and Schrag, 1999; Gaines, 1989)). Effectiveness of knowledge acquisition has also been studied specifically for EXPECT (Kim and Gil, 2000), SOAR (Yost, 1993), and Protégé-2000 (Noy, Grosso and Musen, 2000).

8.1 Quality of questions: cumulative analogy vs. a baseline

A central contribution of the thesis is a demonstration of power of surface-level cumulative analogy for knowledge acquisition. In this section, I present evidence that the performance of the system indeed comes from analogical reasoning from similar objects.

To quantify the benefit of analogy from similar objects, I have conducted an evaluation comparing full analogy as described in Section 4.2 against an ablated (control) version which, instead of formulating questions by mapping properties from *similar* objects O_{src_i}, formulated questions by mapping properties from *random* objects in the knowledge base.

Specifically, the following experiment was performed:

- A *test mode* was introduced into the interface. The test mode was subdivided into two test conditions:
  - **Normal** condition, in which similar topics are chosen as they normally are, and
  - **Random-sources** (control) condition, in which 10 sources for analogy are chosen not based on their similarity, but at random with equal probability among the the objects in the knowledge base. In other words, SELECT-NN

(Figure 4-5) was replaced with a simple stub, but MAP-PROPS (Figure 4-8) and all the consequent filtering mechanisms were left intact.

The contributor was informed that the page presented is in "test mode," but was not told which condition they are seeing. The exact message presented in red on the page was as follows:

> This page has been generated in TEST mode. *The questions generated may differ in quality from the normal mode.* At present, we are injecting TEST mode pages at random to assess what factors affect quality of questions. Please answer the questions as you normally would.

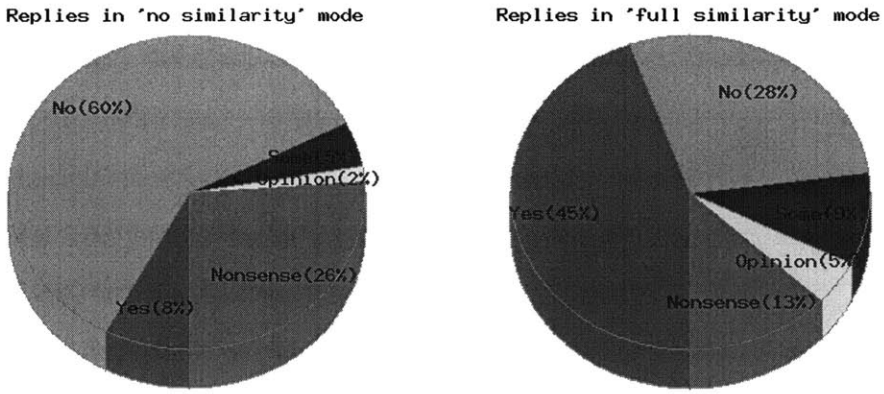
The "test mode" pages were injected at random in place of normal pages with probability 0.3.

- The test condition ("normal" or "random-sources," the user responses, and whether the question was altered was logged (recall that users may alter the question if they wish). Only replies to the unaltered questions were considered. The rates of alteration of questions in either condition were too low to draw statistical conclusions about them.

- The logs were analyzed by taking, for each test condition, the first 1000 replies to unmodified questions and counting the number of questions answered "yes," "no" and so on.

I propose that the system that poses fewer nonsensical questions, as well as does not have a very high rate of "no" answers, with other factors being equal, is better at knowledge acquisition. The reason for valuing "is true" assertions more than "is false" has to do with the fact that, in a sense, there are many more false assertions than true ones. Another way of putting it is that many "nonsense" statements admit the "is false" answer, as will be shown in examples below.

The results are presented in Figure 8-1. The confidence intervals for the values reported were calculated under the assumption that the responses in each condition follow a multinomial distribution. The confidence intervals for any value do not exceed $\pm 3.1\%$, and are smaller than $\pm 2\%$ for values less than 10%.



(a) No similarity used (control).

(b) Full similarity used.

Figure 8-1: Answers to questions generated by analogy from (a) randomly selected and (b) most similar topics.

Of questions generated by analogy, 45% were answered "yes"; the fraction for questions generated from random sources (the control group) was 8%. At the same time, 13% of the questions generated by analogy were ranked as "nonsense," compared with the rate of 26% for the control group. Finally, 28% of those generated by analogy versus 60% of those in the control group were answered "no."

The prevalence of the answer "no" (rather than the answer "nonsense") in the random sources (control) group may seem surprising. In fact, many of the questions answered "no" are very unusual, but can be answered "no." Here are some examples from the data collected in the random-sources condition. All of the following were answered "no":

- "a tea is used to store money"
- "a keyboard is an ear"
- "insects exist to sing"
- "corn is a cigarette"

In all, results indicate that eliminating similarity reduces the quality of the questions posed, shifting an additional 45% of all answered questions into categories "nonsense" or "no."

8.2 Comparison of the resultant knowledge base to the seed knowledge base

In this section, I present the changes in the knowledge base as a result of collecting knowledge with the LEARNER. The analyses presented roughly follow the analyses of the seed knowledge base in Section 7. The figures summarizing the seed knowledge base are reproduced here from Section 7; these figures are presented together with equivalent figures summarizing the resultant knowledge base. Additional comparison by the *kinds* of knowledge collected (classifying the knowledge in the seed and resultant knowledge bases into ontological, meronymical (part-of), and so on) is presented in Section 8.3.

Tables 8.1(a) and 8.1(b) present the gross picture of how the distribution of "is true" assertions in the seed and the resultant knowledge bases.

Figures 8-2(a) and 8-2(b) present, on a log-log scale, the numbers of objects with one, two, and so on properties in the seed and resultant knowledge bases, respectively. They also fit power law curves (which are straight lines in log-log plots) to the data. Note that in the resultant knowledge base, the power has decreased in magnitude from 1.95 to 1.80. One speculation as to the mechanism effecting this change is that more has been learned about "popular" objects — those about which several properties were already known, while very few new objects (which would have few properties) were introduced.

Figures 8-3(a) and 8-3(b) present the average similarity data for seed and resultant knowledge bases, respectively. Refer to Section 7.2 for an explanation. Note that in the resultant knowledge base, the number of objects sharing two properties is at the level expected by chance. One possible interpretation of this data is that the similarity in the resultant knowledge base has been "pushed out" to greater number of properties. That is, instead of a higher than expected number of objects sharing only two properties, similar objects now tend to share three or more properties. LEARNER poses up to 20 knowledge acquisition questions per one acquisition screen. Usually, it can be expected that the contributor will answer many of those affirmatively. A

			Columns (Properties)			
			with ≥ 10 entries	with ≥ 2 entries	with one entry	Total
			291	4905	28070	**32975**
			1%	15%	85%	**100%**
Rows (Objects)	with ≥ 10 entries	723 6%	*2108* *4%*	*9282* *20%*	*17119* *36%*	*26401* *56%*
	with ≥ 2 entries	4277 35%	*4091* *9%*	*15252* *32%*	*23846* *51%*	*39098* *83%*
	with one entry	8049 65%	*1499* *3%*	*3825* *8%*	*4224* *9%*	*8049* *17%*
	Total	**12326** **100%**	*5590* *13%*	*19077* *40%*	*28070* *60%*	**47147** **100%**

#Entries(Assertions)

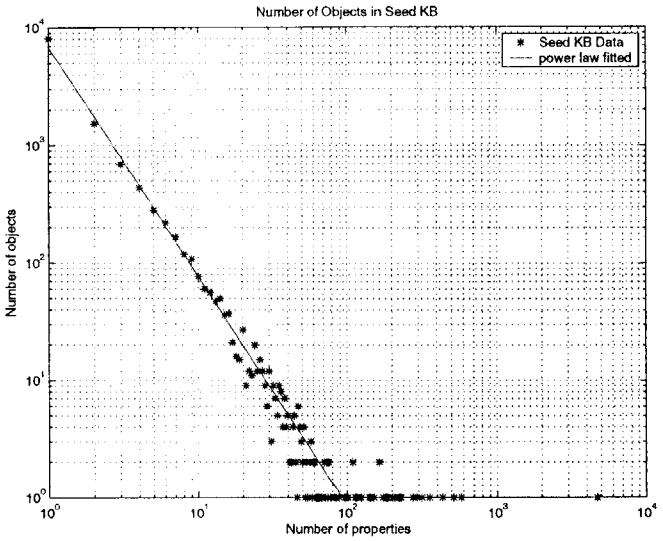
(a) Summary of the seed knowledge base, reproduced from Table 7.1.

			Columns (Properties)			
			with ≥ 10 entries	with ≥ 2 entries	with one entry	Total
			718	7870	30686	**38556**
			2%	20%	80%	**100%**
Rows (Objects)	with ≥ 10 entries	982 8%	*11538* *17%*	*26888* *39%*	*19958* *29%*	*46846* *69%*
	with ≥ 2 entries	4633 36%	*13988* *21%*	*33472* *49%*	*26432* *39%*	*59904* *88%*
	with one entry	8191 64%	*1674* *2%*	*3937* *6%*	*4254* *6%*	*8191* *12%*
	Total	**12824** **100%**	*15662* *23%*	*37409* *55%*	*30686* *45%*	**68095** *100%*

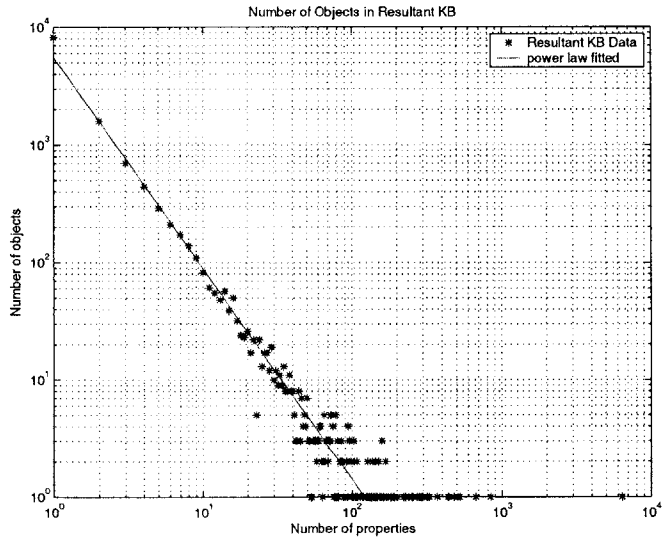
#Entries(Assertions)

(b) Summary of the resultant knowledge base.

Table 8.1: Summaries of the seed and resultant knowledge bases. Total numbers of objects (rows), properties (columns) and entries (assertions) are in bold. Other counts are for rows, columns and entries when only indicated subsets of rows and columns are considered. For example, in the resultant knowledge base there are 982 objects with at least 10 properties, and 46,846 assertions about these objects. For clarity, the "is false" entries are not included in these results.



(a) Seed knowledge base. The solid line is the fitted expression $f(x) = Cx^p$ with $C = 6,789$ and $p = -1.9483$. This figure reproduces Figure 7-1. The values seem to fit Lotka's law.



(b) Resultant knowledge base. The solid line is the fitted expression $f(x) = Cx^p$, with $C = 5,545$ and $p = -1.7957$.

Figure 8-2: Number of objects with N properties in the seed and resultant knowledge bases and a power law fits of the data (log-log plots). In both cases, the power law fit is given by an expression of the form $f(x) = Cx^p$. Values for C and p were chosen by fitting the data for $1 \leq N \leq 50$ to minimize the sum square difference of *logarithms* of real and fitted values.

similar result can be observed when only objects with ten or more properties are evaluated for presence of similar objects, as presented in Figures 8-4(a) and 8-4(b).

Finally, Figures 8-5(a) and 8-5(b) present the percentage of all properties (including unique properties — those that hold for only one object) shared with the nearest neighbors in the seed and the resultant knowledge bases, respectively. The number of properties shared with the nearest neighbor has indeed grown for almost all categories considered, with larger percentage gains for objects with more properties.

8.3 Classes of knowledge acquired

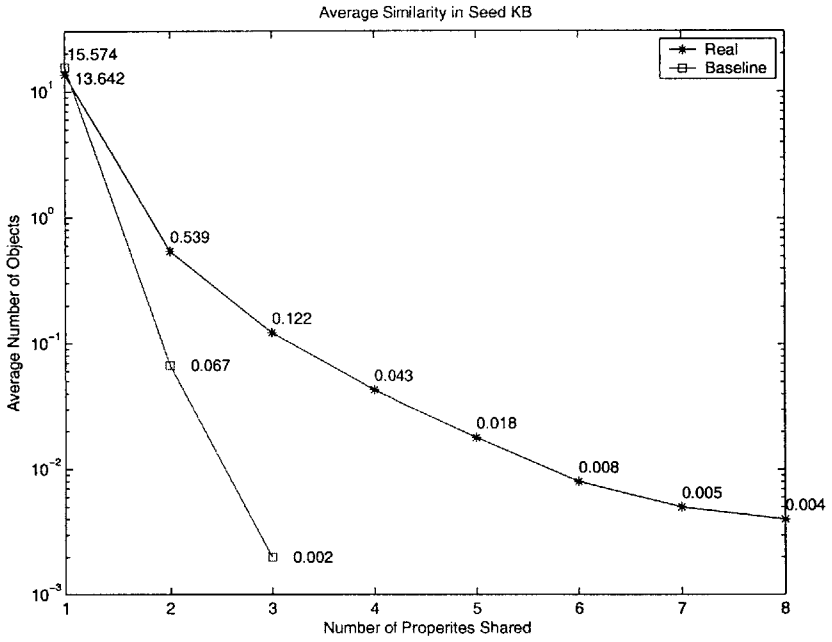
In this section, I examine in greater depth the kinds of knowledge acquired by LEARNER. I introduce a classification scheme for the collected assertions, report on the total and per-class numbers of assertions collected, and, for comparison purposes, report statistics on some existing knowledge bases.

8.3.1 Knowledge classification scheme

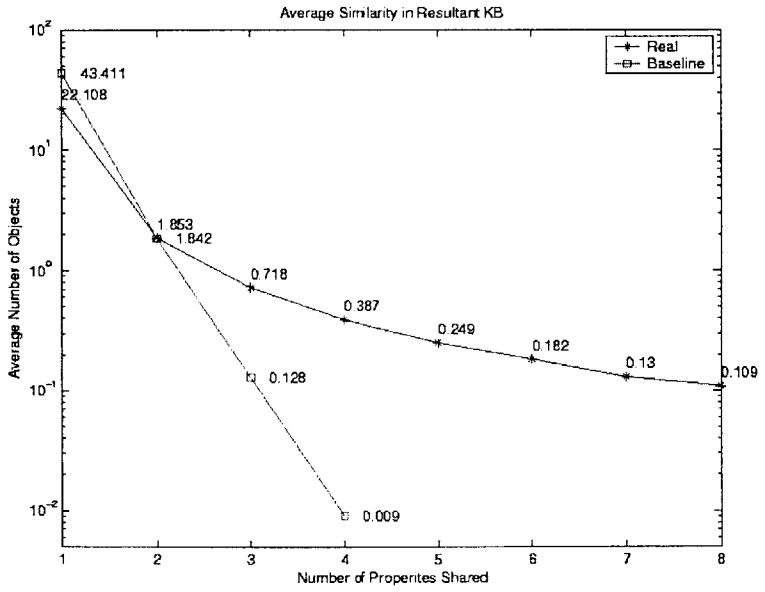
There currently does not seem to be an agreed upon classification scheme for assertions. Some existing approaches to classifying knowledge (together with the data for some knowledge bases) are reported in Section 8.3.3.

I introduce a slightly different classification scheme with thirteen classes, which captures the most common classes of assertions referred to in other literature. The scheme I adopt implemented a set of simple recognizers that has access to the (Link Grammar Parser) parsing of an assertion, parts of speech of individual lexical items, and conjugation information for verbs and nouns. The classes, with examples and features used for classification, are presented in Table 8.2. Each assertion is assigned to exactly one class by the automatic classifier; assertions that are not assignable to any class are placed in the fourteenth, "UNK" (unknown) class.

The fact that the linguistically expressed by contributors knowledge can be classified in these categories deserves further attention. While at the end of Section 3.2 I briefly discuss the kinds of commonsense knowledge that lies beyond stating properties

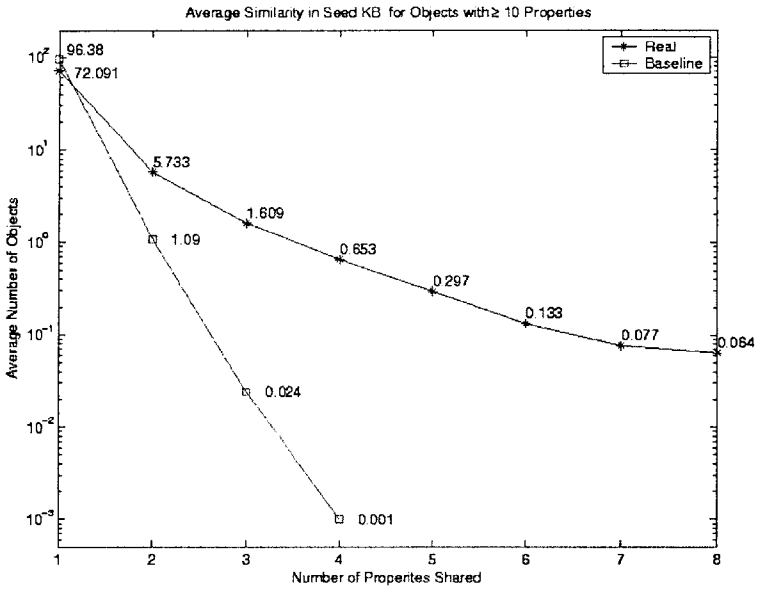


(a) Seed knowledge base. This figure reproduces Figure 7-2.

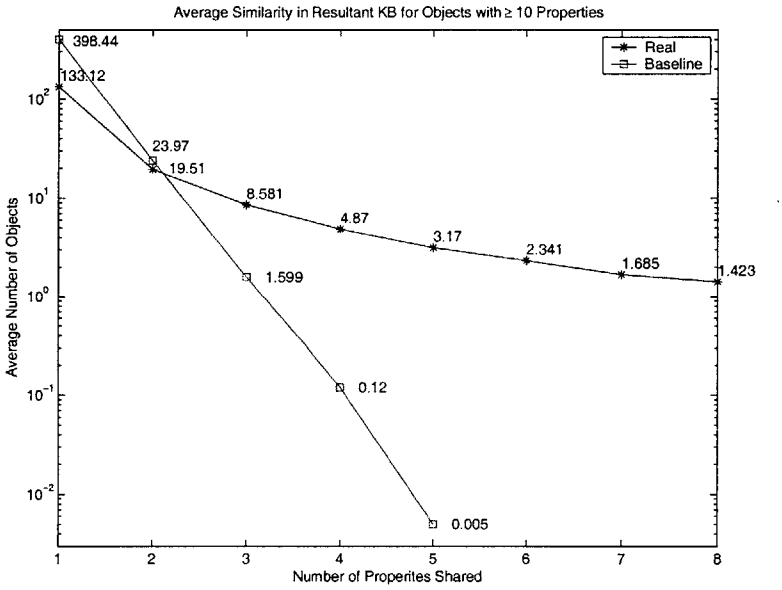


(b) Resultant knowledge base.

Figure 8-3: Average correlation histograms for the (a) seed and (b) resultant knowledge bases. Average amount of correlation in the real and synthetic, (uncorrelated) cases with same frequency distributions are shown. The histograms plots, for different values of N along the X-axis, the number of objects with which a given object is expected to share N properties (on a log-scale Y-axis).

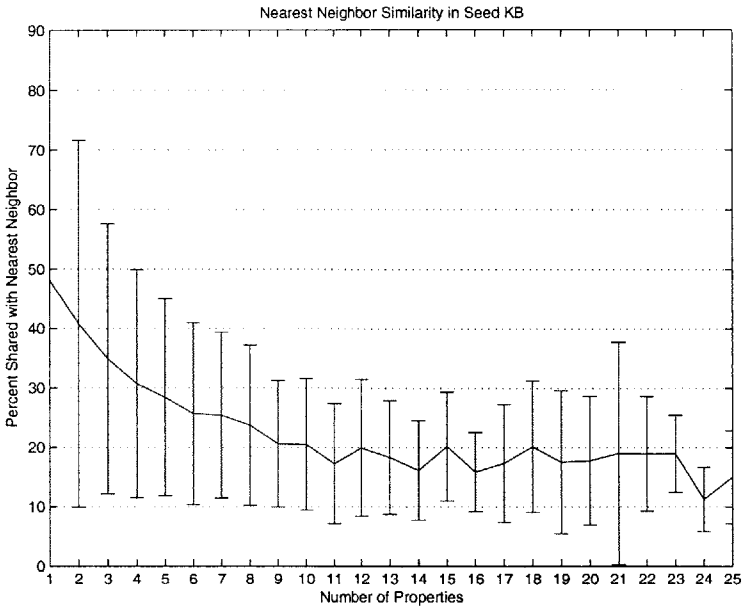


(a) Average correlation histogram for objects with $\geq$ 10 properties in seed KB. Reproduces Figure 7-3.

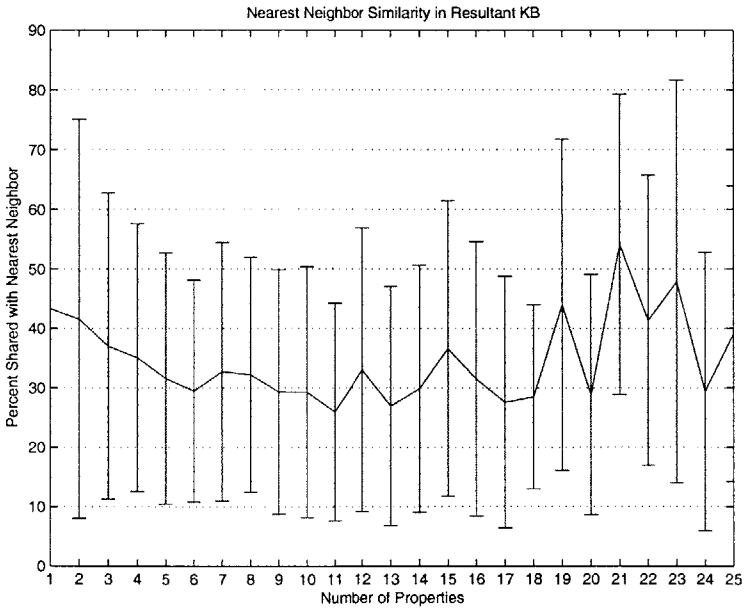


(b) Average correlation histogram for objects with $\geq$ 10 properties in resultant KB.

Figure 8-4: Average correlation histograms for objects in the (a) seed and (b) resultant knowledge bases with $\geq$ 10 properties. Average amount of correlation in the real and synthetic, (uncorrelated) case with same frequency distribution is shown. The histograms plot, for different values of N along the X-axis, the number of objects with which a given object is expected to share N properties (on the log-scale Y-axis).



(a) Seed knowledge base. Reproduces Figure 7-4.



(b) Resultant knowledge base.

Figure 8-5: Nearest neighbor analysis for (a) seed and (b) resultant knowledge bases. For each object, the percentage of "is true" properties shared with its nearest neighbor in the resultant knowledge base has been measured. The X-axis represents the number of properties of an object (results for objects with same number of properties have been averaged), so the values for objects with different numbers of properties can be observed.

Name	Example(s)
ISA	a cat is$_{be}$ an animal$_{noun}$, horses are$_{be}$ quadrupeds$_{noun}$, an armchair *is a kind of* chair.
QUALIFIED-ISA	swans are$_{be}$ white$_{adj}$ birds$_{noun}$.
DEFINITION	phones are$_{be}$ devices (for making calls)$_{prep\text{-}phrase}$.
ACTION	cats eat$_{verb}$ mice.
QUALIFIED-ACTION	kangaroos jump$_{verb}$ (very high)$_{prep\text{-}phrase}$, kangaroos can jump$_{verb}$ (over fences)$_{prep\text{-}phrase}$.
ACTION-ON	horses can be$_{be}$ ridden$_{past\text{-}part}$,
PROPERTY	a swan is$_{be}$ white$_{adj}$, airplanes are$_{be}$ aerodynamic$_{adj}$.
COMPARATIVE	horses are faster *than* people, horses eat more *than* people.
FUNCTION	horses *are used* (for transportation)$_{prep\text{-}phrase}$.
MADE-OF	a window is *made of* glass, audiences *consist of* people.
PART-OF	a wheel is$_{be}$ *part of* a car, a kitchen is$_{be}$ *in* a house.
REQUIRES	writing *requires* literacy, success *requires* extra effort.
POSSIBLE-STATE	horses can be$_{be}$ running$_{pres\text{-}part}$, water can be$_{be}$ boiling$_{pres\text{-}part}$.

Table 8.2: Categories of assertions (with examples). *Italicized* words and syntactic subscripts in examples are cues used by the classifier to classify this example. "be" stands for a form of the verb "be"; "prep-phrase" stands for "prepositional phrase," "past-part" stands for a verb in the "past participle" tense and "pres-part" stands for a verb in the "present participle" tense.

of objects, in this section I presented a categorization of assertions about properties of objects. While many classifications are possible, the above classification maps out some of the territory. This classification scheme suggests, for example, that if contributors are to be queried for particular kinds of knowledge, the contributors should be queried for all of the strongly populated categories to acquire a significant fraction of the possible assertions about objects and their properties.

8.3.2 Knowledge collected

Over a two month period of collecting knowledge, a total of 42,659 assertions were collected. The distribution by the answer received is presented in Table 8.3. In all, 20,315 (47.6%) were "is true" assertions, 10,857 were (25.5%) "is false" and 3,842 (9.0%) were ranked as "nonsensical question."

For comparison, consider that the seed knowledge base contained about 53,447 assertions, of which 96% were "is true" assertions. If only the "is true" and "is false" assertions are considered, the total knowledge base has been grown by 31,172 assertions, or 58.3%, in two months.

Answer	Num Entries	% of Total
Yes	20,315	47.6%
No	10,857	25.5%
Some/Sometimes	5,487	12.9%
Matter of Opinion	2,158	5.1%
Nonsensical Question	3,842	9.0%
Total:	**42,659**	**100.0%**

Table 8.3: Number of assertions collected, by answer received.

The distribution of knowledge by kind of knowledge in both seed and resultant knowledge bases are presented in Table 8.4. The proportions of kinds of knowledge in the seed knowledge base may be indicative of what human contributors spontaneously volunteer, but also reflect the bias of the specific knowledge acquisition templates used in the construction of Open Mind Common Sense(Singh, 2002; Singh, Lin, Mueller, Lim, Perkins and Zhu, 2002), from which the seed knowledge base used in this work has been extracted. Approximately 5.5% of all assertions could not be classified

because they used the highly ambiguous verb "have" to connect the syntactic subject and object of the assertion. By manually disambiguating two hundred cases with the verb "have" in the resultant knowledge base, I estimate that the verb is used to mean "part of," in approximately 60% of the cases. Some examples of such usage are "horses have manes" and "cups have handles." Such assertions were not included in the percentage of "part of" assertions because they were ambiguous. The main verb "have" can also mean "owns" as in "people sometimes have cars", or something else: "children have parents," "every state has a state flower," "a dog can have a litter of puppies," "people have breakfast in the morning."

Overall, the distribution of the acquired knowledge closely tracks that of the knowledge in the original knowledge base. More than one explanation is possible for this phenomenon: the distribution of the knowledge acquired by LEARNER may follow the distribution of the knowledge in the seed knowledge base, or both the acquired and the seed knowledge may reflect a "natural" distribution of assertions that arises in collection from human contributors. Although it is difficult to differentiate without further experimentation, knowledge of how LEARNER operates may favor the former explanation — for example, if the seed knowledge base was heavily skewed towards taxonomic knowledge, the collection process would presumably also pose many questions about taxonomic relationships. If the collection process indeed follows the biases in the seed knowledge base, this property of knowledge acquisition by analogy may be exploited in future work to collect the kinds of knowledge which are most useful.

8.3.3 Other knowledge bases

In this section, I compare the kinds of knowledge collected by LEARNER with the kinds of knowledge aggregated in other knowledge bases.

There are several prior commonsense knowledge bases. The purposes of their construction have varied from primarily enabling machine translation and story understanding (MindNet (Dolan, Vanderwende and Richardson, 1993; Richardson, Vanderwende and Dolan, 1993) and ThoughtTreasure, (Mueller, 2000)) to, more broadly,

Assertion Type	Num in Seed KB	% in Seed KB	Num acqrd	% acqrd	Num in Rslt KB	% in Rslt KB
ACTION	16250	30.4%	9696	30.7%	25946	30.5%
QUALIFIED-ACTION	9680	18.1%	4993	15.8%	14673	17.2%
PROPERTY	5056	9.5%	3202	10.1%	8258	9.7%
UNK*	4809	9.0%	2832	8.9%	7632	9.0%
ISA	4653	8.7%	3025	9.6%	7678	9.0%
ACTION-ON	3372	6.3%	2276	7.2%	5648	6.6%
FUNCTION	2151	4.0%	2080	6.6%	4231	5.0%
PART-OF	1863	3.5%	977	3.1%	2840	3.3%
QUALIFIED-ISA	1748	3.3%	533	1.7%	2281	2.7%
REQUIRES	1656	3.1%	274	0.9%	1930	2.3%
DEFINITION	1005	1.9%	620	2.0%	1625	1.9%
COMPARATIVE	605	1.1%	670	2.1%	1275	1.5%
MADE-OF	407	0.8%	284	0.9%	691	0.8%
POSSIBLE-STATE	192	0.4%	167	0.5%	359	0.4%
TOTAL	**53447**	**100.0%**	**31620**	**100.0%**	**85067**	**100.0%**

Table 8.4: Numbers of assertions by type. First two columns of data present data for the seed knowledge base, middle two for the data acquired by LEARNER and the last two for the resultant knowledge base (seed KB plus the acquired knowledge). 'UNK' represents assertions that could not be automatically classified. Approximately 5.5% of all assertions could not be classified because they used the highly ambiguous verb "have," which means "part of," in approximately 60% of the cases in the resultant knowledge base (e.g. "horses have manes"), but can also mean "owns"(e.g. "people sometimes have cars") or something else (e.g. "children have parents," "every state has a state flower").

being a linguistic database (WordNet (Fellbaum, 1998; Miller, 1998)) and enabling reasoning via resolution theorem proving (CYC and OpenCyc, (Guha and Lenat, 1994; OpenCyc, 2001)). The methods of construction for the knowledge bases have varied from hand-coding by experts (e.g. WordNet, ThoughtTreasure, CYC) to automatic extraction from machine-readable dictionaries (MindNet).

Constructing a sufficiently large knowledge base is by no means a simple endeavor, and even the largest of these are by no means complete (the difficulty of authoring a large knowledge base is one of the motivations for deploying the LEARNER system).

Two tables published by Mueller (Mueller, 1999, 2002) on some existing knowledge bases have been reproduced here. Table 8.5 presents, for different knowledge bases, the number of concepts, the number of a-kind-of and is-a assertions, the number of part-of or material-of assertions, and the number of other assertions.

Name	Concepts	ako/isa	part-of/ material-of	Other
Cyc*	149,052	97,172	16,000+	1,497,000
Cyc Upper Ontology 2.1 (Cycorp, 1997)	2,846	7,161	0	2,579
Mikrokosmos (Mahesh, 1996)	4,500	-	-	-
MindNet (Richardson et al., 1993, p. 9)	45,000	47,000	14,100	32,900
SENSUS (Knight and Luk, 1994)	70,000	-	-	-
ThoughtTreasure 0.00022 (Mueller, 2000)	27,093	28,818	666	21,821
WordNet 1.6 (Fellbaum, 1998; Miller, 1998)	99,642	78,446	19,441	42,700

Table 8.5: Number of concepts and common relations in other knowledge bases. Note that relations in MindNet, being automatically extracted from dictionary text, are not always correct. Adapted (with permission) from Mueller (1999). *Data about the CYC system has been obtained separately and describes KB 616 version of the CYC knowledge base as of January 2003 (personal communication, K. Panton, 2003). In addition to 97,172 "genls" assertions (expressing, e.g., that all apples are fruit), CYC also contains 435,622 "isa" assertions (expressing such assertions as "United States is a country" and "Father-of is a two-place predicate") The "other" figure includes the "isa" assertions, but does not include additional 288,450 "bookkeeping" assertions stating such meta information as who and when added the knowledge.

Another currently growing knowledge base is the Open Mind Common Sense (OMCS), from which the seed knowledge for LEARNER has been extracted. OMCS,

Type of relation	OpenCyc 0.6.0, 2002-04-03	ThoughtTreasure 0.00022,1999-12-08	Explanation
hierarchical	37,755 (62.0%)	28,818 (56.2%)	ISA, collections
typing	20,293 (33.3%)	845 (1.6%)	argument types, selectional restrictions
other	1,246 (2.0%)	16,518 (32.2%)	other
definition	1,005 (1.7%)	25 (0.0%)	concept definitions, equivalences
implies	569 (0.9%)	0 (0.0%)	logical implications
part	10 (0.0%)	807 (1.6%)	object parts and substances
object property	0 (0.0%)	422 (0.8%)	object properties
spatial	0 (0.0%)	1,796 (3.5%)	typical locations, arrangements on grids
script	0 (0.0%)	2,074 (4.0%)	relating to scripts
Total	**60,878 (100.0%)**	**51,305 (100.0%)**	–

Table 8.6: Knowledge in OpenCyc and ThoughtTreasure. Adapted with permission from Mueller (2002).

like LEARNER, does not disambiguate the assertions it collects. Furthermore, OMCS does not provide any syntactic or semantic filters on contributor input, and does not collect truth values. OMCS, however, has wider coverage — its scope includes collection of stories, descriptions of images, and so on.

The data summarizing OMCS as of early 2002 is reproduced in Table 8.7. As of January 13th, 2003 the knowledge base had 494,489 contributions, with identical or similar assertions sometimes contributed more than once (personal communication, P. Singh, 2003).

Overall, it can be seen that CYC currently has a significant lead over all other efforts. Furthermore, the knowledge in CYC is fully disambiguated (its concept hierarchy is finer-grained than WordNet's and was specifically designed to capture world rather than lexical knowledge). However, the approaches of collecting knowledge from volunteer contributors (both OMCS and LEARNER) have collected (admittedly ambiguous) knowledge at a much lower development cost and shorter time frame (for example, OMCS has been collecting knowledge for two years rather than CYC's nearly two decades). Significant promise for future growth both for CYC (personal communication, D. Lenat, 2003) and for other (freely available) large scale common-

Class of Knowledge	% of Collected
Scripts/Plans	14.4
Causal/Functional	11.9
Spatial/Location	10.2
Goals/Likes/Dislikes	5.5
Grammatical	5.5
Photo descriptions	5.4
Properties of people	4.8
Explanations	2.6
Story events	1.8
Other	37.9
Total	100.0

Table 8.7: Knowledge in OMCS. Adapted with permission from Singh (2002).

sense efforts may lie in relying on volunteer contributors, together with developing methods for disambiguation and acquisition of (highly) unambiguous knowledge from contributors who have had very little training.

8.4 Rate of contribution to Learner

In this section, I address the behavior of visitors to the site, examining the volume of contributions, frequency of return visits, and so on. The top four sources from which contributors arrive at the LEARNER ("1001 Questions") web site are as follows:

- MIT Artificial Intelligence Laboratory Projects page[1] (about *40%* of the contributors),
- MSN search engine and directory (about *25%*),
- Google (about *12%*),
- Gamespotter[2] (about *8%*).

The remaining 15% comes from other sources. The terms used in the search engines to find the site range from "free online game" and "online knowledge games," to "questions games" and "game making questions." Some queries to Google are

[1] http://www.ai.mit.edu/research/projects/projects.shtml
[2] http://www.gamespotter.com

"open mind 1001" and "1001 Questions game," suggesting that the searches are specifically for the LEARNER web site.

To make using LEARNER less of a chore, LEARNER does not require contributors to log in or identify themselves. The only information collected about a contribution is the IP address of the computer from which the contribution was made (as reported by the client to the web server on which LEARNER resides), the exact time at which a given contribution was made (i.e., the time when a assertion was passed to LEARNER from the web server), and a unique session id generated at the beginning of the session and stored on the contributor's machine. Cookies on the client machine are used to store information about the previous topic the contributor was teaching LEARNER about, and to provide a unique identifier for each contribution session (for the purpose of later statistical analysis of the contribution behavior). Cookies do not survive for more than one hour from the time of last contribution in a session.

The IP address of the contributing computer should be considered to be at most a crude proxy for the identity of the contributor. For example, the one-to-one correspondence between contributors is violated when computers reside behind the firewall of a large company. All such computers may appear to have the same IP address, making contributors from different machines appear to have the same IP address. A single physical computer in a home may be shared by family members, computers in a school computer lab may be shared by many people affiliated with the school, and a computer in an Internet cafe (Internet cafes currently being the dominant mode of Internet access in many countries) may be shared between many customers.

Conversely, a single contributor may be using a machine with a dynamic IP address, may be sometimes visiting LEARNER from work and sometimes from home, or may be accessing the Internet via Internet cafes, using a computer with a different IP address each time.

Despite the above cautions about interpreting IP addresses as unique contributor identifiers, an analysis of the total number of assertions contributed per IP address can still provide some valuable insights. Such an analysis is presented in Figure 8-6. For the purpose of this discussion, a contribution is any assertion (parsable or

not, already present in the knowledge base or not, and with any truth value). The logarithmic Y axis represents the number of contributions. Note that the period for which this data has been tabulated is 72 days rather than 61 days over which figures for Section 8.3.2 were calculated. The total number of contributing IP addresses over the 72 days is 1047. The average number of contributions per IP address is 59.2.

One notable trend in Figure 8-6 is the plateau (a relatively large number of IP addresses) with exactly twenty contributions. The responsible factor may be the fact that the system presented up to 20 knowledge acquisition questions per screen. A large portion of the IP addresses making twenty (or fewer!) contributions, presumably saw one set of questions of the system, answered those questions, and have not proceeded further.

Another aspect of contributor behavior is the number of their repeat visits to the system. One possible way to estimate the number of repeat visits to the system would by looking at the number of unique "sessions," where a single session is tracked by a cookie set on the client computer. Because cookies are sometimes turned off on the machines of the contributors, cookies have not proven very useful in tracking the identity of a session. Instead, the present analysis defines a notion of a *contributing day* — for a given IP address, a 24 hour period (starting at midnight US Eastern time) is called a contributing day if a contribution of at least one assertion (with any truth value) was made during this 24 hour period from this IP address.

The analysis of the number of contributing days for 1047 contributors over 72 days is presented in Table 8.4. Note that the count does not exactly represent how many times a contributor decided to have a session with LEARNER. The approach of counting "contributing days" collapses genuine multiple visits per day into a single "contributing day," causing an underestimation of the number of visits. On the other hand, a single session that continues through midnight of US Eastern time results in two "contributing days." Presumably, the numbers of "contributing days" per IP would be larger if calculated over longer period, especially because some contributors had their first visit to the site at the end of the period and have not yet had a repeat visit by the end of the measurement period. On a contributing day, the average

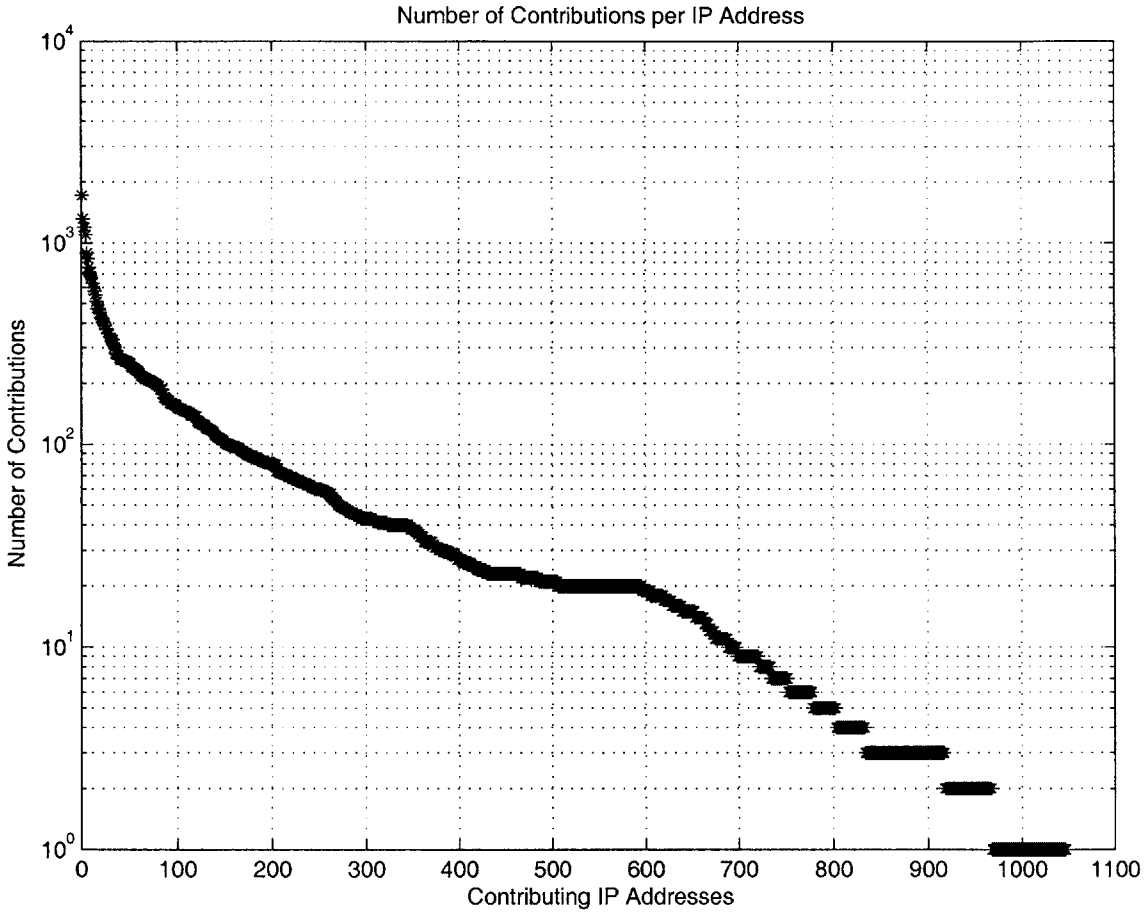


Figure 8-6: Number of contributions per IP address. Distinct IP addresses are listed along the X axis. The actual addresses have been suppressed, their rank is indicated instead. The addresses are arranged in the order of decreasing number of assertions contributed. A contribution is any assertion (parsable or not, already present in the knowledge base or not, and with any truth value). The logarithmic Y axis indicates the number of contributions.

contribution from a single IP address was 54.29 assertions.

Number of Contributing Days	Number of such IP Addresses
1	987
2	47
3	3
4	2
5	3
6	2
7	2
8	0
9	1
Total	1047

Table 8.8: Number of IP addresses that had N contributing days, calculated over initial seventy two days.

While both numbers on the number of contributions per IP and the repeat visit rate per IP may be somewhat difficult to interpret, the provided data may be useful in comparing LEARNER in its current form to a different system that makes similar data available. This data can also be useful in calculating the effect on the contribution volume and repeat visit rate of such additional factors as interface improvements, introduction of prizes for best contributors, or emailing subscribing contributors with a newsletter summarizing the progress of the site.

8.5 User feedback about Learner

One of the goals of this work is to create a large knowledge base of commonsense assertions. In order for this effort to succeed, in addition to technical competence of the knowledge collection process, it is desirable to ensure that visitors to the site have a positive experience. To find out what the contributor perceptions are, I have asked them to comment on the system.

For the period of fifty days, every page of questions presented by LEARNER contained the following instructions (together with a text entry box for the comments):

The site is in beta testing. We are extremely eager to receive your feedback on which features of the site you find helpful or annoying, and what you can suggest by way of improvement.
Enter your comments here. Comments on quality of questions posed and ease or difficulty of answering them (with examples as appropriate), are particularly needed. Include your email address if you'd like.

In all, 118 complete, non-redundant comments were received. Mainly, the comments asked for a clarification of documentation or more information. Many comments also pointed out manifestations of three bugs in the system, which have since been addressed. There were on the order of 30 comments that have commented on the quality of the overall site, the interface, or suggested specific extensions of the current functionality.

The remaining comments have focused on the quality of the site and the concept it embodies or have requested new features.

Of the sixteen comments on quality of the site, twelve (75%) were positive, for example:

> "Congratulations, I shall access this wonderful project often even though I use a public computer and have little time" - From Sao Paulo, Brazil,
>
> "This is a very interesting system. I am very fascinated. I will give that link to all my friends,"
>
> "This is a very interesting experiment,"

one was mixed:

> "Your program is quite smart, but it asks too many nonsensical questions,"

and three were negative, for example:

> "Needs to work on the grammar."

The feedback that focused on the interface generally expressed that the interface was not very intuitive at first, but that the contributor eventually figured it out. One contributor praised the transparency of system's function:

> "It's very useful having all the '[i]' buttons [because] you can find out easily how to drive it."

A few users, while happy with the overall experience, have pointed out the syntactical limitations of the system, for example:

> "It *never* accepts my mathematical info,"
>
> "Not being able to parse lists is cumbersome,"
>
> "Faster data entry method would be nice; maybe have an 'expert user' mode as an option."

The few remaining comments requested an array of features. One of the most popular suggestions (four comments) was to extend the acquisition strategy to validate the knowledge collected. The users have commented that both typos and human error in entering the knowledge can contribute to errors, in particular to false assertions being stored in the system as true.

For example, one comment read:

> "Eventually you need a way of changing established facts caused by typos."

Two contributors have suggested adding the answer

> "I (personally) don't know"

to the set of available choices. See Section 5.2 for a discussion of this.

Finally, contributors have suggested that more advanced reasoning by the system would further improve the quality of the system's questions. Contributors have cited scenarios where (i) the ability to *generalize* from specific answers and (ii) ability to carry out rule-based inference would have been helpful.

One concrete example of need for generalization quoted was: when presented with a set of assertions "fire can heat X," "fire can heat Y," "fire can heat Z," the system should be able to generalize that "fire can heat any object."

The need for rule-based reasoning was motivated by the example: once "X is not a food" then the system should not ask whether "X is tasty."

To sum up, it seems that contributors were largely satisfied with the site, provided useful information on the bugs that were consequently eliminated. The critical comments have suggested features that are mostly in line with what I perceive to be the limitations of the current system (Section 8.5.1) and are discussed in the Section 9.2 (Future Work).

8.5.1 Limitations of cumulative analogy

The simplicity of LEARNER's fundamental algorithm for posing questions brings into high relief the additional types of reasoning that are currently missing from the system.

Given the success criterion that a knowledge acquisition system should collect useful knowledge, a question-posing knowledge acquisition system can fall short of our expectations in two ways: (i) not posing the questions which would be useful (failing to collect some useful knowledge), and (ii) posing questions which are not useful (collecting useless, often redundant) knowledge.

Cases where the system poses questions that are not useful are easier to spot, and I address them first.

Based on my own experience with the system and the contributors' feedback, the following currently missing types of reasoning can be identified. Implementing these should lead to an improvement in question quality:

Handling implications. For example, the system currently has no way to establish that "if something is neither edible nor drinkable, one should not enquire whether it is tasty." This is because the system cannot *explicitly* represent that "tasty" applies only to edible or drinkable things (here, scoping is treated as

a kind of implication). The inability of cumulative analogy to handle implications seems mostly acceptable when acquiring abstract assertions. In a domain where a lot of causal or configuration-based reasoning was required, this limitation would be a lot more pronounced.

Handling disjointedness. The system currently does not understand that some situations are mutually exclusive with others. For example, knowing that "people walk on 2 legs" does not prevent the system from asking (for example, by similarity with dogs or cats) whether "people walk on 4 legs."[3]

The questions that would be useful but were not posed are revealed indirectly: some questions posed by the system reveal that it should have instead posed *different* questions, as follows:

Handling generalization. The system sometimes asks numerous overly specific questions. For example it may pose the question "X exists in three dimensions?" about many objects in place of X. The reasoning mechanism needed to avoid asking these given the more general assertion "Objects exist in three dimensions" is already implemented in the current system (see Section 4.3.1). What is missing is the ability to *acquire* the more general knowledge without it being volunteered.[4]

Additionally, the analysis of the seed knowledge base in Chapter 7 has revealed that the seed knowledge base has a large number of unique assertions (a large number of properties each of which was asserted about a single object). While some of the unique properties are quite exotic (such as the property of a "caterpillar" which states "`will turn into a butterfly`"), the inability of reasoning by analogy directly over properties to acquire new properties is an important limitation of cumulative analogy. I feel that allowing the contributor to both add new knowledge and to modify the

[3]Note, however,that knowing what is disjoint is non-trivial. For example, English and French are two different languages, but it does not mean that a person that speaks one does not speak the other.

[4]Note that to handle generalization in the syntactic object position, inferring quantification of syntactic objects would also have to be addressed.

knowledge acquisition questions when answering them ameliorates this shortcoming in LEARNER as it has been deployed. Section 9.2 (Future Work) further elaborates on the above points and discusses how they can be addressed within the framework of cumulative analogy.

Chapter 9

Discussion

This chapter consists of three sections. In the first section, I overview some of the relevant prior work from the expert system, machine learning, knowledge acquisition interface, and text mining traditions. There is a vast body of literature in each of these areas. In overviewing the literature, I focus on the work that has been particularly influential to my thinking regarding knowledge acquisition. The second section discusses future work regarding the knowledge representation and the algorithms that, in light of the experience with the deployed system, are likely to improve the quality of the knowledge acquisition questions. Also briefly discussed are ways to improve the resulting knowledge base. Finally, and perhaps most importantly, the third section summarizes the most important contributions of the completed work.

9.1 Background

Prior work that is relevant to this work can be categorized as follows:

1. Early expert systems.

2. Systems that form expectations from prior data and use the expectations to aid knowledge acquisition.

3. Knowledge representation, especially seeking to capture what is easily expressed in natural language.

4. Machine learning, especially in concept formation and learning association rules.

5. Natural Language Processing, especially extracting relationships from free text (text mining) and answering questions by finding answers in text (question answering).

6. Projects gathering (and using in an application) knowledge from ordinary web surfers who elect to participate.

These categories span several research communities, but they all provide relevant background for some aspect of this thesis. I examine the work in each category in turn, explaining its relevance to the theory and the system developed in this work. First, however, I review the literature on the amount of commonsense knowledge that needs to be collected to capture a rough equivalent of the commonsense knowledge a ten year old child might have.

9.1.1 Amount of commonsense knowledge

When addressing the task of collecting commonsense knowledge, a natural question that arises is how much of commonsense knowledge do humans possess. Answering this question would permit bounding the task of collecting commonsense knowledge and estimate, however crudely, the accomplished progress towards the goal of acquiring the commonsense knowledge of a human.

Furthermore, estimating the amount of a certain kind of knowledge or estimating the amount of knowledge in a given subdomain of commonsense can allow one to estimate when the sufficient amount of such knowledge has been collected and when can acquisition move on to a different kind of knowledge or a different subdomain.

However, addressing even the fundamental question of the total amount of commonsense knowledge is difficult. In my opinion, answering this question is complicated by two properties of human knowledge. One problem is the question of granularity of human knowledge, and the related question of where the line between commonsense and esoteric expert knowledge should be drawn.

The hierarchical (fractal) structure of knowledge at various levels of specificity can well be observed in medical literature. According to estimates and calculations reported by S. G. Pauker and Schwartz (1976, pp. 14–15), an average person without medical training knows approximately 100,000 real-world facts that are relevant to medicine. In addition to this, two popular (and roughly equivalent) textbooks on general internal medicine have approximately 2,000 pages and have approximately 100 facts per page. At the next level of specificity, about ten subspecialty texts covering areas such as nephrology, cardiology and hematology can be identified, each with textbooks containing approximately 60,000 facts each (S. G. Pauker and Schwartz, 1976, pp. 14–15). Even accounting for overlap between subfields, the authors of the investigation believe that there are roughly twice as many new facts (i.e. 400,000) introduced at this level of specificity. Any binary division of facts into "common" and "uncommon," it seems, would have to introduce an arbitrary boundary.

The second problem with estimating the amount of human knowledge is the inferential nature of the knowledge. For example, Lenat cautions against adhering too closely to numbers of terms and axioms in an inferential knowledge base, referring to both numbers as being a "red herring." Lenat's critique of excessive focus on the numbers of axioms has to do with the inferential nature of the knowledge base — several axioms can compactly express something that is equivalently (or almost equivalently) expressed by thousands of axioms. One example is that "any animal belongs to at most one species" can be expressed either as the ontology of species and a single statement asserting that the leaves of the ontology are disjoint, or as a (much larger) set of assertions each stating disjointedness of a given pair of species (Lenat, 1995, p. 35).

Despite these difficulties, there are several approaches to roughly estimating the amount of commonsense knowledge a ten year old or adult human living in a modern society possesses.

One early discussion is due to Minsky:

> "My impression, for what it is worth, is that one can find fewer than ten areas, each with more than ten thousand "links." One can't find a hundred

> things that he knows a thousand things about. Or a thousand things, each with a full hundred new links. I therefore feel that a machine will need to acquire on the order of a hundred thousand elements of knowledge in order to behave with reasonable sensibility in ordinary situations. A million, if properly organized, should be enough for a very great intelligence. If my argument does not convince you, multiply by ten" (Minsky, 1968, pp. 25–26).

In publications predating the "red herring" stance, Lenat et al. approximate the amount of knowledge necessary for human-level commonsense reasoning at 10^8 axioms, a number three orders of magnitude higher than the lowest of Minsky's earlier figures (Lenat, Guha, Pittman, Pratt and Shepherd, 1990).[1]

Some more specific psycholinguistic data that sheds light on the issue of the total amount of commonsense knowledge have been aggregated in the work by Dahlgren et al. (1989) on naive semantics. To assess which features humans typically associate with concepts, the following psycholinguistic studies have been conducted: To arrive at the most "characteristic" features of categories such as DOG, LEMON, and SECRETARY, between 20 and 75 subjects were asked in earlier studies to freely list "characteristic" features of these categories. Features that were freelisted by at least one fifth of the subjects were chosen for another experiment, in which subjects were asked to rate the typicality of the features. The number of features ranked as "highly typical" in the second experiment averaged 15 per topic (this description has been adapted from Dahlgren et al. (1989, pp. 153–154)). If one assumes the passive vocabulary of concepts to be on the order of 100,000 (which exceeds an average person's linguistic vocabulary), the total number of such agreed upon characteristic features is on the order of several million.

Perhaps the most convincing bound on how much knowledge humans retain comes from the studies of how much information humans are able to remember and retrieve after a long period. Landauer has carried out a review and quantitative analysis of his own and others' experiments on human ability to memorize various types of informa-

[1] Unfortunately, I have been unable to discover the detailed origins or derivation of this estimate.

tion. The analysis included memorization of visual, verbal, and musical information and has yielded similar results for the rate at which information can be remembered so that it can be retained over the long term. All estimates point to humans being able to retain approximately two bits per second (Landauer, 1986). This implies a per-year rate of only about 8 megabytes, or several hundred megabytes over a lifetime. These estimates, of course, do not include any non-learned information, such as the information present in the "hardware" humans use — for instance, it does not include any hard-wiring for causal or spatial reasoning that may be encoded in the genome. Still, if the figure for what human brains are able to retain is in the hundreds of megabytes within an order of magnitude, the overall project of collecting even tens of millions of axioms should be achievable (provided that the right knowledge is being collected!) by collecting from tens of thousands of contributors in merely a few years.

9.1.2 Early expert systems

A great deal of knowledge acquisition work happened in the late 80's and early 90's in the context of expert systems. Some tools (BLIP, ILROID, INDUCE, ID3) constructed expert systems by running a non-interactive procedure on a large data set to induce a set of rules. Other tools (EXPECT, FIS, KREME, MEDKAT, NEXPERT) constructed expert systems by relying on a knowledge engineer to elicit knowledge from an expert and encode that knowledge.

These shells generally either hard-coded for a particular problem solving method of the performance application (MOLE, SALT) or the domain (OPAL, STUDENT), had to settle for gathering only shallow knowledge, or required extensive programming by an AI programmer for each new problem. A good overview of the approaches taken by earlier KA systems is (Gaines and Shaw, 1992).

More recently, much work has focused on moving towards "multifunctional knowledge bases" — systems that can support more than one problem-solving approach (Aamodt, 1995). This trend correlates with increased attention to acquiring and encoding *problem solving methods* (PSMs). PSMs are a kind of meta knowledge, specifying how other, factual knowledge should be used. The idea has been that by

having a large library of problem solving methods, task independent factual knowledge can be processed by applying the right methods to it. There has also been some work to simplify the acquisition of PSMs (Kim and Gil, 1999). Two overview papers (Boose, 1989; Menzies, 1998) contain a detailed classification of and further pointers to many implemented systems.

9.1.3 Forming expectations from existing knowledge

Using previously acquired knowledge to aid future acquisition is a major theme of this work. Here is an overview of existing systems which leverage knowledge that they have in order to guide their consequent interviewing of experts:

- TEIRESIAS (Davis, 1979) worked with experts to help maintain a knowledge base. The system helped acquire and refine proposed rules by analogy with similar rules that were already known. For example, if similar rules typically included an additional constraint about the patient's age, it would suggest that the new rule may need such a constraint as well.

- SALT (Marcus and McDermott, 1989) is a system for helping solve design problems, such as selecting elevator doors, cables, and motor type given (i) the requirements of a particular building and (ii) lookup tables of costs of various parts. The system was able to examine the set of rules the expert gave it for how they "fit together." If there were some inconsistencies or if not enough information was present to run the system, it would ask the expert to fix its rules. It was hard-coded to find places were rules could be fixed and suggest those to the expert.

- EXPECT Method Developer (EMeD) (Kim and Gil, 1999) helps an expert contributor create problem-solving methods for the EXPECT system. The system works by analyzing interdependencies between methods to create expectations about what other methods need to be defined and what they will look like. Because methods have stated capabilities which are much like return types of

functions in a programming language, and the capabilities are organized in a hierarchy, the system can presuppose that methods with similar capabilities will have similar requirements (inputs). The system also checks if a method uses yet undefined types and proposes to either change it to an existing type or define the type at a later time.

An approach to eliciting new knowledge that has enjoyed a lot of popularity because of its simplicity is based on Kelly's Personal Construct Psychology (PCP) (Gaines and Shaw, 1993). It uses repertory grids to rank various constructs (properties) of several elements (objects) in a grid.

For example, a grid can be made with the American presidents as the columns of a grid and their "party affiliation," "popularity," and "charisma" as rows of the grid.[2] Each president than has a numeric value on each of these scales. The elicitation technique associated with the approach is to compute which elements seem similar or identical based on the data known so far and then ask for a new construct (property) that would distinguish the two. Another surprisingly effective method (called triad-based elicitation) is to select any three elements and query what two elements have in common that a third does not.

While somewhat constrained in what *kinds* of knowledge can be acquired, the PCP-based approaches are effective in gathering a lot of knowledge rapidly. The knowledge gathered in a grid can also be analyzed in a couple of interesting ways: the INDUCT (Gaines, 1993, p.465) algorithm infers subsumption rules from the grids and the FOCUS algorithm (Shaw, 1980) clusters the elements in the grid in a hierarchy by their similarity. (The work on the COBWEB and CLASSIT algorithms has subsequently built upon the FOCUS algorithm.)

While the knowledge in LEARNER is more heterogeneous than simple numerical values on scales of the PCP-based approaches, an interesting potential extension of my work is to apply the LEARNER-collected knowledge the same methods of processing as to the knowledge collected with PCP-based approaches. Such possible

[2]At the time of writing, WebGrid III, a system that allows construction and analysis of both simple and advanced grids, was available online at http://gigi.cpsc.ucalgary.ca.

post-processing includes induction of rules and extraction of hierarchies from (clustering of) the objects and their features.

Reasoning by analogy, similarly to case-based reasoning, can be viewed as a form of leveraging prior knowledge in understanding new examples. Analogy has proven to be a powerful tool when knowledge is acquired incrementally.

Particularly relevant to this approach is Winston's work on analogy (Winston, 1972) which focused attention on the differences new knowledge has from what is already known, and, in consequent work, processed English-like input to be able to draw analogous inferences given similar new input (Winston, 1982).

Also relevant are the notions of analogy in TEIRESIAS (Davis, 1979), and Forbus's approach to analogy by mapping between partially aligned structures of concepts in two domains, as formalized in the work on the Structure Mapping Engine (SME) (Forbus, Falkenhainer and Gentner, 1986).

9.1.4 Knowledge representation

I briefly overview several systems that are aimed at representing knowledge. LEARNER uses structures storing simplified parsed natural language as a representation, but also has the ability to recognize assertions made in natural language into a frame-like representation.

Early work such as the OWL language for knowledge representation (Szolovits, Hawkinson and Martin, 1977) was strongly based on natural language, and pursued the idea that the meaning of any term came from all its uses in the knowledge base rather than from a formal definition or axiomatization (the view later built on by work on naive semantics, Dahlgren et al. (1989)).

The most ambitious project has been CYC, a multi-decade endeavor to encode commonsense knowledge by a team of enterers (Lenat, 1995). The project has evolved as it unfolded, currently representing knowledge in CycL, a formal logic-like language. CYC comes with its own browser, inference engine and some ability to process and generate natural language.

Other systems aimed at capturing the sort of knowledge this work tries to capture

also take a logic-based approach. The most notable examples are the KL-ONE family of systems, including CLASSIC (Brachman, McGuiness, Patel-Schneider and Resnick, 1990).

Another ongoing project with the goal of creating an intelligent agent capable of communicating in natural language is SNePS (Shapiro, 2000).

Another recent development elaborating on well-established approaches is KARL, a language incorporating frame-logic (Fensel et al., 1998), and DAML+OIL, a knowledge representation language expressed in RDF (DAML, 2002). These systems establish a baseline for what is expressible and inferable in today's systems, as well as exemplify the existing approaches to representing different kinds of knowledge.

9.1.5 Machine learning: concepts and relationships

Machine learning provides techniques for extracting concepts and rules from data.

Algorithms such as FOIL (Quinlan and Cameron-Jones, 1995) and CHILLIN (Zelle, Mooney and Konvisser, 1994) learn from positive and negative examples to form Horn clauses describing these, in effect creating descriptions for classes of objects from examples. While the simpler ones try to reflect the given data exactly, the more elaborate work (Brunk and Pazzani, 1991) introduces information-theoretic stopping criteria to avoid overfitting noisy data. For more details on inducing rules from the observed data, a good overview can be found in (Califf, 1998).

Some work that aims to bring machine learning to other aspects of knowledge acquisition includes FOCL-1-2-3 (Brunk and Pazzani, 1992). This system automatically generates hypotheses and allows an expert to select the correct ones from the generated ones. This system also maintains connections between rules and the examples they explain, making knowledge maintenance easier.

Some work has also looked at interleaving machine learning and knowledge acquisition to make knowledge acquisition easier (Sommer, Morik, Andre and Uszynski, 1994; Webb, Wells and Zheng, 1999; Morik, Wrobel, Kietz and Ende, 1993). However, current systems still construct knowledge acquisition interfaces for contributors that need to be trained in using these tools. In contrast, this work takes a different

approach -- enabling collection in plain English from untrained contributors.

9.1.6 NLP: text mining and question answering

The advent of the web has made large textual corpora readily available, leading to an explosion of work the fields of text mining and question answering.

Text mining work has especially focused on extracting ontologies (taxonomic information) from various sources, as overviewed in (Maedche and Staab, 2001).

While LEARNER does not analyze text written for other people, it is engaged in extracting knowledge from the mass of contributed assertions (to pose new knowledge acquisition questions). Hence, some approaches in text mining may be useful on this task as well. Some particularly relevant work in the field includes (Faure and Nédellec, 1998; Hahn and Schnattinger, 1998; Maedche and Staab, 2000).

The advent of the web also brought about a resurgence in work on question-answering systems (Cardie, Ng, Pierce and Buckley, 2000; Dahlgren, Ljungberg and Ohlund, 1991; Kwok, Etzioni and Weld, 2001).

Typically, these work in conjunction with a search engine to try to retrieve from collection of documents, not just a relevant document, but a specific answer to a given question. These systems generally lack domain knowledge and rely on language processing and statistics, rather than deduction, to find the right answer. While these systems are typically built to aid question-answering, they can be viewed as knowledge extraction systems.

Additionally, systems such as MindNet (Dolan, Vanderwende and Richardson, 1993; Richardson, Vanderwende and Dolan, 1993) and that of Hearst (Hearst, 1992) process textual corpora or machine readable dictionaries to extract not answers to questions, but "is-a" (and, in case of MindNet, other) relationships between the concepts present in the text being analyzed.

All of the above systems face many of the same challenges that LEARNER does: the need to canonicalize text, the ability to cope with incorrect or out-of-context assertions, and the ability to weigh evidence for and against something being true.

9.1.7 Gathering from contributors over the web

Most people in the world are not trained in knowledge representation. Thus, collecting knowledge from these people presents a challenge. Also, there are many people in the world, and therefore it is desirable to be able to acquire from many contributors, pooling their contributions and bootstrapping on their individual expertises.

The knowledge acquisition community has typically approached this challenge by designing user interfaces that allow a user to enter and query knowledge by filling in forms. Protégé-2000 (Li, Shilane, Noy and Musen, 2000) is a project seeking to break out this functionality in a modular fashion, but numerous other user interfaces were created, including the WebGrid system based on personal construct psychology (Gaines and Shaw, 1998), the OntoEdit browser, CYC browser, EMeD, and so on.

While these approaches do simplify knowledge entry, they do so by constraining what knowledge can be entered and by forcing the contributor to use a particular way of representing knowledge (sometimes changing the meaning subtly). Because the contributor has to be skilled in the particular representation employed, these approaches are more suited to simplifying the entry for trained contributors.

On the Internet, there are also several simple systems implemented for entertainment purposes. These are, effectively, knowledge acquisition tools tied to performance applications:

- Guessmaster.com is a Web site containing several games that pose previously gathered questions trying to guess the person, object, animal, TV show, or movie (depending on the game). Data gathered is in public domain from John Comeau.

- 20Q.net, http://www.20q.org is a learning program that plays the game of 20 questions. It gathers and re-uses the information useful in guessing an object. It is an interesting demonstration of gathering from a community on a specific, well-defined "toy" problem.

- Open Mind *Animals*, is forthcoming online at

http://openmind.org/Animals.html (Stork and Lam, 2000).

Similarly to the above systems, it a learning program that tries to guess the animal a human player is thinking of by posing yes/no questions.

When these systems fail to guess the object the player is thinking of based on the player's answers, these systems ask the player to contribute a new yes/no question that could help differentiate the player's object from others.

Out of machine learning tradition comes the Open Mind Initiative (Hearst, Hunson and Stork, 1999), an umbrella project with the explicit goal of gathering a variety of knowledge from ordinary "netizens." The Open Mind Initiative is quickly gathering support; it focuses on creating a common platform of tools for gathering from netizens, sharing the collected data, cross-validating their input, rewarding the best contributors, and so on.

To date, one of the most ambitious projects to gather commonsense knowledge from untrained contributors is Open Mind Commonsense (Singh, 2002), which uses templates and prompting for free-form text to gather knowledge.

9.2 Future Work

The goals of this work have been to define a clear vision — to acquire commonsense knowledge from untrained contributors, and to take concrete steps towards this vision — to formulate, implement, deploy and analyze knowledge acquisition via cumulative analogy. To maintain focus, many fascinating and promising directions had to be left outside of the scope of this work. Here are some of these, with the topics I believe to be most important being higher on the list:

Richer internal representation. The knowledge is currently stored in a form that is close to how it was entered. Introducing the ability to reify more abstract assertions derived from the entered assertions, together with an update of how questions are actually posed, should allow for better knowledge acquisition. The

new capability can be introduced without significant changes to the question-formulating or filtering algorithms. For example, recognizing that "sugar is sweet" as asserting "has a (specific value of) taste" and "Tabasco sauce is spicy" as stating that "has a (specific value of) taste" could lead the system to infer by the current mechanism that, for example "`salt`" or "`pepper`" also have specific values of taste, leading the system to ask what that value is (i.e. ask how salt or pepper taste). This would allow LEARNER to tap related, not only identical properties in posing its questions.

Additional reasoning mechanisms. The system could be outfitted with additional mechanisms both (i) to filter the presented knowledge (causing LEARNER to pose fewer non-useful questions) and (ii) to formulate questions by means other than cumulative analogy, working side by side with cumulative analogy (causing LEARNER to pose additional useful questions). These are discussed in greater detail in Section 9.2.1.

Better use of collected assertions. The assertions in the knowledge base are currently used in two ways: (i) to establish similarity between objects in Select-NN and (ii) to create new hypotheses for knowledge acquisition in Map-Props. As is discussed in Chapter 7, there may be significant number of correlations in properties between objects that are not necessarily similar (for example, many metal objects, similar or not, tend to be hard and shiny). The presented algorithms could be extended to extract correlations between properties of non-similar objects and to use these correlations to pose new knowledge acquisition questions. This is elaborated in Section 9.2.2.

Better estimation of importance of properties. Currently, importance of each property is computed according to functions *Wt* and *FreqWt* (see Section 4.2). These rely on the number of objects for which a property is true to estimate how significant it is. Indeed, very common properties (e.g. "a person can hold X" is true for very many objects X) probably tell us little about object similarity. However, there are many exceptions to this general trend. For example, there

are many "animals" (and hence assertions "X is an animal," and yet being told that something is an animal is quite informative. It would be interesting to estimate importance of a property by how much information it provides. Perhaps this could be estimated from how many correlations or implications can be derived from asserting this property. If a more sophisticated and yet tractable algorithm for computing "importance" can be introduced, properties with high importance could be made to have a greater effect on similarity than the unimportant ones, resulting in improved similarity judgments and presumably better questions.

Additional critics. For example, the system could refrain from asking questions which are very similar to each other, or questions whose answers depend on other questions being posed at the same time. Critics could also employ some additional method completely orthogonal to the generation methods to assess quality of the generated questions. As a simple example, they can use statistical plausibility of word pair co-occurrence in the question to remove the implausible questions (or to lower their scores).

Better use of collected replies. Currently, only "Yes" and "No" replies affect the operation of the system in a sophisticated way. The other answers are merely stored; the other answers only prevent the identical question from being posed again. Other collected replies could be used better. Possible strategies specific to particular replies are discussed in Section 9.2.3.

Other work, while not addressing question quality, can further improve the quality of the resulting knowledge base. Two main directions for improvement are as follows:

Improving reliability. Currently, a limitation of the system as a whole is that an incorrect assertion cannot be overridden or corrected. Empowering contributors with the ability to override something the system already believes would be a path towards making the knowledge in the knowledge base more reliable. For example, according to one contributor's comment, the contributor has added this

assertion as a result of accidentally omitting an '*n*': "lilies are *ice*." Two subtasks can be identified: (i) automatically detecting "suspicious" assertions and (ii) designing a mechanism that would allow contributors to override incorrect contributions while preventing a malicious editor from significantly corrupting the existing knowledge base.

Removing ambiguity. The other way to improve the knowledge base is to remove some of the ambiguity present in the contributed assertions. It can be removed, for example, by having contributors disambiguate some of the lexical items (words) in the assertions into their WordNet senses, in a manner similar to (Chklovski and Mihalcea, 2002), possibly in conjunction with unsupervised disambiguation methods. Specific proposals on addressing the ambiguity are discussed in greater detail in Chapter 6.

As has been discussed in Chapter 1, LEARNER can be viewed as a system that leverages human contributors to clean up "noisy" hypotheses that it constructs. As discussed in Section 9.1.6, other research efforts such as MindNet and Hearst's hyponym acquisition system (Hearst, 1992), generate assertions (noisily) from a different source — by mining machine readable dictionaries or arbitrary texts from the World Wide Web. Hence, perhaps a larger system that combines text mining and acquisition and verification from human contributors, is possible. This approach could be particularly appealing for acquiring seed knowledge in new subdomains.

9.2.1 Additional reasoning mechanisms

The two most important reasoning mechanisms to implement would be rule-based reasoning and generalization from examples. Rule-based reasoning could be used in a number or ways:

- to suggest new questions — if the inference is probabilistic, one could ask whether assertions inferred with medium or low confidence are true;
- to filter questions — for example, not asking about the taste of something that is not edible or drinkable

- to improve evaluation of similarity of two objects — by comparing not only what is known about them explicitly, but also what can be inferred about them;

A typical rule could be represented as a conjunction of left-hand-side assertions (preconditions) and a right-hand-side assertion (implication). The terms could be expressed (as well as indexed and retrieved) as assertions, with the additional possibility of turning the base form of any word into a variable. For example, "?cats? eat meat *because* ?cats? are carnivores." Here, "?cats?" represents a variable that can be substituted with other terms (the original term is preserved to provide information on syntactic conjugation of the new term being substituted. The standard forward and backward chaining methods can be run over these rules to perform inference.

Necessary for rule-based reasoning are the rules themselves; I briefly discuss how they can be acquired. There are two sources: the rules can either be mined from the knowledge base, or acquired directly from contributors. Mining them from the knowledge base is discussed in Section 9.2.2.

As for acquiring rules directly from contributors, one strategy would be for the system to watch for the assertions that receive the answer "no" and request the reason for it being so. For example, upon learning that "cows do not eat meat," the system could enquire for the reason (the antecedent of the rule), and receive as the reason "cows are not carnivores." From that, variablizing the matching components in the antecedent and the consequent (in this case "cows"), the system would acquire the rule:

$$\neg A(X, \textit{be carnivore}) \Rightarrow \neg A(X, \textit{eat meat}).$$

The rule may be refined later in light of additional information (for example, the above rule may be applicable only when X is an animal).

The other significant reasoning mechanism suggested by my and the contributors' experience with the system is generalization (inductive inference) from examples. The idea is to use assertions about the more specific concepts in a hierarchy to acquire assertions about the more general concepts.

One approach to implementing generalization would be as follows. At any node in the taxonomy of all known entities (which is a directed acyclic graph with a designated top node), an assertion can be said to hold for "all," "none," or "some" of the subsumed nodes. For example, the assertion "animals need to breathe" would be interpreted as the property "need to breathe" having the value "all" at the node "`animal.`" Similarly, "some living things need sunshine" would be interpreted as "some" at the node "`living thing.`" For most nodes, the value for a given property would be "unknown." An algorithm could then be defined to hypothesize values for higher nodes based on the values of subsumed nodes. For example, if a property has the value "all" for all known direct subnodes of a node, one can hypothesize that it also has the value "all" for the node itself. Some preliminary experiments indicate that a more relaxed policy (for example, at least three positive disjoint examples and no counterexamples) should also suffice for fairly robust generalization.

9.2.2 Better use of collected assertions

Clusters of properties may exist on a sub-object level, and cut across object similarities. Mining such correlations between properties, with attention to taxonomic relationships, should be possible.

The task of mining correlations between properties is quite similar to the standard task of mining association rules in datamining (Agrawal, Imielinski and Swami, 1993). Particularly relevant is the work that also accounts for taxonomic relationships between objects (see, for example, (Han and Fu, 1995)). In datamining, however, a particular datapoint is either present or unknown. In contrast, in this task, each datapoint can, at a minimum, be in one of three categories: known to be true, known to be false, or unknown. The evaluation of correlation (in particular, the notions of strength and support of an association rule used widely in datamining) would have to be extended to reflect the richer input in this particular task.

Using a datamining-like analysis of the data, certain pairs or larger groups of strongly correlated properties could be uncovered. The resulting correlated groups can be used in a number of ways:

- to hypothesize and acquire rules for use in a rule-based inference mechanism,
- to directly pose knowledge acquisition questions (without ever resorting to variablized rules),
- by looking for perfectly correlated pairs of properties with sufficient support, redundant properties (multiple ways of phrasing essentially the same thing) can be identified and the redundancy removed from future KA questions and during evaluation of object similarity,
- make question posing based on similarity between clusters of properties (note that this might make the system less transparent).

9.2.3 Better use of collected answers

As mentioned above, only the answers "Yes" and "No" have a sophisticated impact on the questions posed by LEARNER. The system could be extended to make better use of the other replies currently being collected .

For example, an assertion being answered with "Some/Sometimes" (e.g. "hats protect heads from blows,") could be used as a sign to ask a more specific question. That is, the system could ask what kinds of hats that protect heads from blows, to learn, for example, that a "`hard hat`" and a "`helmet`" are the such kinds of hat.

Once a contributor ranks an assertion as "Nonsensical" or "Matter of opinion," the system could try to develop a mini-theory (even if a very primitive one) of why this is so. In many cases, for example, the nonsensicality is due to the object belonging to a class of objects that do not combine with this property. For example, upon learning that "beaches do many things with money" is nonsensical, the system could not only avoid posing this question about "`beach`" in the future, but also decrease the score of this question for things similar to "`beach`."

9.3 Contributions

This work makes the following contributions:

1. **Theoretical and empirical demonstration of the power of reasoning by shallow semantic analogy (namely, cumulative analogy) for knowledge acquisition.** Cumulative analogy is an N nearest neighbor technique that maps the properties that hold about nearest neighbors onto the target. In this work, it is used to formulate the knowledge acquisition questions.

   The implementation of cumulative analogy, detailed in Chapter 4, consists of two algorithms: **Select-NN**, which selects the nearest neighbors O_{src_i} to the current topic O_{target}, and **Map-Props**, which maps properties from these nearest neighbors onto O_{target}. The output of Map-Props is used to form knowledge acquisition questions.

   The theoretical analysis of cumulative analogy in Chapter 7 established an upper bound on what fraction of assertions are *directly* acquirable (acquirable without the bootstrapping effect of acquired knowledge enabling additional acquisition). Direct reachability is a lower bound of the asymptotic reachability realized if the acquisition process is allowed to bootstrap from the knowledge it acquires.

   On the seed knowledge base used, I showed that reachability ranges between 47.3% and 67% for different strictnesses and scopes of reachability. Note that the analysis (and the percentages reported) are restricted to the objects with sufficient number of properties, with the threshold for the minimum number of properties known varying between 2 and 20.

   I have also empirically demonstrated that basing the analogical reasoning on the nearest neighbors is responsible for the observed success rate of posing knowledge acquisition questions by analogy. Evaluating the percentages of questions answered affirmatively, negatively and judged to be nonsensical in the cumulative analogy from nearest neighbors case compares favorably with the baseline, no-similarity case that draws analogies from randomly selected objects in the knowledge base rather than on nearest neighbors. Of the questions generated by cumulative analogy, contributors answered 45% affirmatively, 28% negatively and marked 13% as nonsensical; in the control, no-similarity case 8% of ques-

tions were answered affirmatively, 60% negatively and 26% were marked as nonsensical.

The application of cumulative analogy to knowledge acquisition has also uncovered some limitations of the approach. I identify the three most significant ones. The first is its need for a (preferably large) set of "seed" knowledge to base analogies on. The second is its inability to acquire properties not shared with other objects in the knowledge base. This is ameliorated in the implemented system by mixing knowledge acquisition by analogy with allowing the contributor to modify the acquisition assertions and to volunteer additional knowledge. The third is its unawareness of taxonomic "is a" relationships between objects when such relationships are present.

Lack of ability to reason about taxonomic relationships leads analogical reasoning alone to pose many redundant questions. For example, for each particular kind of animal, the algorithm may ask if this animal "`can breathe.`" Simply establishing that "animals can breathe" and being aware of the taxonomic relationships should have been sufficient. This limitation has been addressed with the additional module that filters questions which are inferable from more general assertions.

2. **Introduction and characterization of cumulative analogy, a class of analogical reasoning algorithms which pool evidence from multiple nearest neighbors.** This class of algorithms operates on a collection of assertions about objects having and lacking properties. Although applied to knowledge acquisition, these algorithms exhibit a set of features that I believe makes them an attractive reasoning method for a wider set of tasks, such as querying the knowledge base by analogy for question answering and checking consistency of the knowledge base.

   The operation of cumulative analogy and its characterization have been addressed in the previous point. The chief strengths of this class of algorithms are as follows:

Ability to bootstrap. The amount of bootstrapping has also not been quantified, but it is also easy to observe. One example of bootstrapping (also given in Section 4.2.5) is as follows: when starting with the seed knowledge base and teaching about "`newspapers`," the similar topics, together with their similarity scores, are: "`book`" (6.72), "`map`" (2.76) "`magazine`" (2.67), and "`bag`" (2.51). The three highest-scoring knowledge acquisition questions posed are "newspapers contain information?", "all newspapers have pages?" and "newspapers are for reading?" If these questions are answered affirmatively and the answers are submitted to the system, the set of the similar objects remains the same, but their scores become: "`book`" (10.94), "`map`" (5.53) "`magazine`" (4.12), and "`bag`" (2.51). As can be seen from the change in similarity scores, the less similar topic ("bag") became less influential in creating knowledge acquisition questions relative to others. This should lead to questions posed by LEARNER being more focused. Conversely, when a question is answered negatively, the source topic(s) from which this question was formulated will become less similar on future iterations, again helping LEARNER suppress irrelevant source topics.

Ability to cope with noise. Although the ability to cope with noise has not been quantified empirically, it has been observed a number of times and should be easy to reproduce. One example of this, given in Section 4.2.5, has to do with acquiring knowledge about "`tool`" starting from the seed knowledge base. Although the set of similar objects includes such spurious matches as "`fire`" and "`horseshoe`", the knowledge acquisition questions posed by cumulative analogy are still quite reasonable. As is discussed in Section 4.2.5, as long as the total level of noise is relatively low, noise tolerance allows the system to cope with noise arising from insufficient knowledge, lexical and structural ambiguity of the collected knowledge, and noise arising from accidental or malicious incorrect submissions by the contributors.

The measure of reach of analogy is closely related to measures of correlatedness of knowledge also introduced by this work and summarized below.

3. **Novel measures of correlatedness of knowledge.** Chapter 7 introduced two measures correlatedness between objects in a collection of assertions about objects and their properties.

   Measuring how correlated objects in a knowledge base are (i.e. how many properties or what percentage of properties they share) may have additional applications, but in this work it has been important in analyzing applicability of analogical reasoning. If pairs of objects that share more properties than can be expected by chance are rare, analogical reasoning about objects based on matches of properties may be irrelevant to such a knowledge base. If, for any object, there is at least one other object that shares very many properties with it, one method of analogical reasoning may be appropriate. If, instead, there are many objects that share with it exactly two (rather than many) properties, another method may be more appropriate.

   The first measure introduced in this work is the "average similarity histogram." It calculates with how many objects, on average, a given object in the knowledge base shares one property, with how many it shares two, and so on. This measure has been used both on the real knowledge base as the starting point for knowledge acquisition by analogy, and on a synthetic knowledge base with the same frequency characteristics of properties, but with correlations between objects due purely to chance.

   It was used to establish the difference between the observed amount of similarity with what could be expected by chance, and to provide an indication of where on the spectrum between rare strong correlations and frequent weak correlations the real knowledge base is located. The results indicated that a mixture of both few strongly and many weakly correlated objects are present; these results motivate the chosen approach to analogical reasoning, with several (up to ten) most similar objects contributing to similarity, with those most

strongly correlated having a bigger impact.

One limitation of the average similarity histogram approach is that it averages data for objects with potentially vastly different numbers of properties. By doing so, it loses information on what percentage of all properties on an object were shared. To bring clarity to this issue, a different measure was introduced: percent similarity of an object to the object most similar to it (in other words, to its nearest neighbor). In this work, the average data and the standard error was computed for twenty-four categories of objects: the first category contained all objects with two properties, second with three, and so on. Grouping objects by categories in this way allowed us to see how the percentage of properties shared behaves with respect to the number of properties of an object. The analysis uncovered that objects in most categories shared about 20% of properties with their nearest neighbor, with objects with fewer properties sharing more than 20%. Note that the denominator in the percentage includes all properties, even those unique to a single object in the count (i.e. those that could not be shared with the nearest neighbor).

The results of applying this measure contributed to my choice of using more than a single nearest neighbor as the source of mapping in Map-Props, the algorithm that generates properties likely to be true. The expected coverage that would result from using many nearest neighbors has been bounded with another measure, the "reach of analogy," which is described above, in the discussion of the first contribution.

I conjecture that these measures can be applied to other knowledge bases to compare the applicability of knowledge acquisition by analogy between knowledge bases. These measures could be used to assess whether and what form of analogical reasoning is appropriate to a given knowledge base.

4. **A simplified object-property representation for assertions in natural language.** A popular view in the Artificial Intelligence community maintains that the choice of representation is frequently more important than the choice of

algorithm. Formulating a representation for natural language assertions which facilitates comparison of properties for equality of meaning has been an important part of this work.

I have introduced the "signature" representation, over which all comparison of properties for equality is carried out. The signature of a sentence is meant to preserve its most important information; it is the set of nouns, verbs, adjectives (together with their parts of speech) that appear in subject, main verb, object(s) and prepositional phrases of a sentence, as well as adverbs in adverbial phrases. All words that make up a signature are reduced to their basic forms. For example, the assertion "all dogs bark" has the signature: "{dog_{noun}, bark_{verb}}." Signatures are discussed in greater detail in Section 3.3.

I believe that the signature representation plays an important role in performance of Select-NN, the algorithm that establishes the nearest neighbors. Informal experimentation in the exploratory stages of this research looked at alternative schemes that retain more information from a sentence (for example, retains the base form of all the words present in the sentence). The experimentation suggested that matches between properties in these more detailed representations were in practice often insufficiently frequent for Select-NN to return a strong set of similar objects.

5. **Demonstration of applicability of the generate-and-filter architecture to the task of generating knowledge acquisition questions.** Generate and test is a classic, well known approach in the Artificial Intelligence community. The contribution this work makes is the demonstration of its applicability to the particular problem of generating knowledge acquisition questions. I speculate that the generate and test architecture has applied well to the task of formulating knowledge acquisition questions due to two characteristics of the task: (i) the lack of constraint on the methods for formulating questions, and (ii) presence of constraints on what questions should not be posed. Two examples of the latter include not posing questions that the system will not accept,

and not posing questions the answer to which is already known.

This work also contributes to the scientific community a growing knowledge base of assertions, a resource that will be useful in implementing practical systems that use common sense. The knowledge base should also enable a variety of semantic approaches to processing natural language.

Additionally, I hope that the public launch to a wide audience of a system that uses the state of the art in language and knowledge processing promotes the fields of artificial intelligence and natural language processing by attracting attention of a broader audience to the open problems. In other words, I hope this work contributes to popularizing the state of the art and the current challenges in the fields of AI (specifically, knowledge representation and analogical reasoning) and NLP (specifically parsing, language generation, and ambiguity in language).

This work has also taken steps in characterizing the kinds of commonsense knowledge that need to be acquired in order to capture human-like common sense. In this work, I have taken a "naive semantics" view of commonsense knowledge. In Section 3.2, I briefly described the kinds of knowledge that lie outside the set of assertions about objects and their properties. In Section 8.3.1, I have presented a more fine grained classification of assertions about properties of objects. A further elaboration of this effort should be useful in organizing the field's pursuit of capturing and representing commonsense knowledge, which should some day enable commonsense reasoning by machines.

Lastly, a contribution this work makes to further research is a simple tool that allows experimentation with a variety of approaches to reasoning over natural language. The thoroughly commented source code of the LEARNER system is available under an open source license at:

http://sourceforge.net/projects/learner .

It includes a number of features not exploited by LEARNER itself but documented in the manual included with the distribution. The features include support for variables, rules, and rule-based inference, both over natural language assertions and their

signatures, and over a frame-like internal representation of knowledge.

Appendix A

Link Grammar Parser

The *Link Grammar Parser* is a constraint-based English-language parser that tries to assign a consistent set of *linkages* between all words in a sentence.

The Link Grammar Parser is an impressive system in its own right. The parser comes out of CMU, is written in C and its source code is freely available for non-commercial purposes.

Complete distribution and extensive documentation of the link grammar parser was available at the time of writing at http://www.link.cs.cmu.edu/link.

For readers unfamiliar with the parser, here is brief example of how the parser would parse the sentence "`cats eat mice`":

```
   +-Sp-+--Op-+
   |    |     |
cats.n eat mice.n
```

As a list of links, this information can be represented as follows (the number following each word indicates the position of the word in the sentence):

```
((cats.n,0 Sp eat,1)
 (eat,1 Op mice.n,2))
```

The above parsing contains the following information about the word "cats":

- "`cats`" is a *noun* — because of "`cats.n`",

- "cats" has the subject role in the sentence – it is on the left side of an "S" link.
- "cats" and "eat" are *plural* — they are linked by the "Sp" link in which the lowercase "p" denotes plurality.

The parser has also been equipped with a post-processor that can pull out constituent tree structure (noun phrases, verb phrases, and so on) from the linkage data. However, LEARNER uses the parser's native representation because it holds more information and is better suited for analyzing similarity.

Appendix B

FramerD

FramerD is a distributed object-oriented database authored by Ken Haase and used by LEARNER. FramerD is available under the LGPL and includes persistent storage and indexing facilities that can scale to very large database sizes, as well as a language *FDscript*, a superset of Scheme.

LEARNER is implemented in FDscript.

FramerD also comes with a version of the WordNet lexical database and a released part of the CYC ontology combined and converted into the FramerD format (the database is called BRICOLAGE or BRICO). LEARNER uses the WordNet component only.

FramerD also has many attractive features:

- Built-in support for distributed operation.
- FDscript, a Scheme-like scripting language well suited for AI-type applications.
- Built-in functions for XML and HTML parsing and output.
- Built-in support for perl-like regular expression pattern matching.
- Built-in support for nondeterminism.

FramerD documentation, covering the database implementation and the FDscript language, was available at the time of writing at http://framerd.org.

Appendix C

Natural Language Generation

This work addresses knowledge acquisition from contributors not trained in knowledge engineering or formal logic. To acquire knowledge from such contributors, this work performs knowledge acquisition in natural language. Using natural language removes the need to train contributors in a knowledge representation formalism, but it also creates two additional requirements on the system: (i) addressing lexical and structural ambiguity in the collected knowledge, and (ii) generating acceptable natural language.

The effects of ambiguity in the language on the algorithm itself as well as possible ways of disambiguating the collected knowledge have been discussed in Chapter 6. This appendix describes the processing implemented in support of the latter issue, that of natural language generation.

Specifically, LEARNER carries out the following processing to formulate grammatically correct questions:

Tokenization. When a newly added assertion is being processed, tokens such as "fire engine" and "time travel" are identified in it by comparing against the compound nouns and verbs listed in the WordNet lexical database. Identified tokens are treated as single, indivisible words just as a normal single word.

Noun conjugation. LEARNER formulates questions about both plural and singular nouns, which requires the capacity to use either the singular or plural form of

a noun. For example, when mapping the assertion "books contain knowledge" onto "`newspaper`," LEARNER needs to use the plural of "`newspaper`" to formulate the assertion "newspapers contain knowledge." The number of the word being substituted is determined from the information provided by the parser about the original sentence.

In LEARNER, the correct form is obtained from a lookup table for irregular nouns and according to standard spelling rules for the remaining nouns (Jones, 2002) — for example, "box/boxes," "baby/babies," and "shelf/shelves."

Verb conjugation. Processing assertions containing negations sometimes also requires conjugating the main verb of an assertion. For example, consider that

> "a cat does not *have* wings," is transformed internally into
> "a cat *has* wings" (with a truth value of 0).

The need for conjugation arises because in sentences with an auxiliary verb the auxiliary verb carries the information about tense and agreement with the subject (while the main verb in such sentences is in infinitive form), while in sentences without an auxiliary verb it is the main verb that carries the number and tense information. Once a negation is removed from a sentence containing an expression such as "do not" or "does not," so is the auxiliary verb "do" or "does," and the main verb must be conjugated to agree with the subject and also to carry the information about the tense of the assertion.

In LEARNER, the conjugation of irregular verbs is carried out by retrieving the proper form in a lookup table, and conjugation of other verbs is carried out according to standard conjugation rules (Jones, 2002) — for example, "walk/walks," "carry/carries," and "garnish/garnishes."

Injecting and removing indefinite articles. There are cases when replacing one noun with another — something that is routinely done by LEARNER — requires injecting or removing an indefinite article associated with that noun.

The reason for this lies in the fact that nouns in English fall into two classes: countable and non-countable. Some examples of non-countable nouns (also called *mass nouns*) are: "salt," "soil," "software."

While singular countable nouns require an indefinite article (for example, "*a* carrot is tasty"), mass nouns do not (for example, "salt is tasty"). LEARNER determines when an indefinite article should and should not be used by looking up whether a noun is in a list of known mass nouns, and adjusting the assertions it outputs accordingly.

Selecting the correct indefinite article. A rather cosmetic feature of LEARNER is to select "a" or "an" as appropriate. The agreement of the article with the following it word is cosmetic because the agreement is not enforced by the underlying Link Grammar parser (so, for example, both "a cat has a tail" and "an cat has an tail" are considered valid), and the article is never included in the signature. However, early experiments with LEARNER indicated that contributors were confused by improper article being used (a situation that can arise, for example, when the countable noun following the indefinite article changes from one that starts with a consonant to one that start with a vowel). An example of the need to adjust the article is when "`rhino`" is replaced with "`elephant`" in the following assertion: "*a* rhino has ears." The assertion needs to be changed to "*an* elephant has ears."[1]

In the implemented LEARNER system, every question that is about to be posed is processed and the correct indefinite article is selected for each noun that needs an indefinite article. Ignorance of this approach to which words were changed is intentional; it allows for independence of the article agreement module from any prior processing, making the system more modular.

[1] Also note that whether "a" or "an" is used depends on the word following it, not on the noun it modifies: "*a* male elephant has tusks."

Appendix D

Deriving the Amount of Correlation Due to Chance

This appendix derives a formula for $E_{av}(O,k)$, the expected number of objects in a knowledge base with which a randomly selected object in the knowledge base will share k properties with. This expression is used to derive the baseline in Figure 7-2 analytically, without performing a simulation.

Before proceeding with the derivation, some notation needs to be introduced. Let U denote the total number (the "universe") of distinct properties in the knowledge base. Let O_i denote an object with exactly i properties, and N_i denote the number of objects with exactly i properties (only "is true" properties are considered in this analysis). I also use the standard combinatorics notation $\binom{n}{m}$ to denote $\frac{n!}{m!(n-m)!}$.

Lemma D.1 *Given a non-negative integer k s.t. $k \leq U$, suppose that an object O_i has i properties ($k \leq i \leq U$) which have been randomly selected with equal probability and without replacement form a universe of U properties. Further suppose that an object O_j has j properties ($k \leq j \leq U$) selected from U in the same manner. Then, the probability that O_i and O_j share exactly k properties (for $k \leq i, k \leq j$) is given by any of the three equivalent expressions:*

$$\frac{\binom{i}{k}\binom{U-i}{j-k}}{\binom{U}{j}} = \frac{\binom{j}{k}\binom{U-j}{i-k}}{\binom{U}{i}} = \frac{i!\,j!\,(U-i)!\,(U-j)!}{U!\,(U-i-j+k)!\,(i-k)!\,(j-k)!\,k!} \tag{D.1}$$

Proof:

This can be established by a simple counting argument. Since all assignments of properties to O_i and O_j are assumed to be equally likely, the probability of the event of interest (O_i and O_j sharing exactly k properties) can be computed by dividing the number of events in which the condition of interest is realized by the total number of events being considered. The first expression above can be established by assuming some i properties of O_i have been chosen and counting the number of ways properties of O_j can be chosen.

For a given selection of i properties out of U for O_i, the number of ways that O_i and O_j can share k properties is the number of ways k "shared" properties can be chosen out of i multiplied by the number of ways the $j-k$ non-shared properties of O_j can be chosen from $U-i$ properties that are the complement of properties of O_i. This, independently of which i of U properties O_i has, gives rise to the numerator $\binom{i}{k}\binom{U-i}{j-k}$ in the first expression above. The total number of ways j properties can be chosen out of U is $\binom{U}{j}$, giving rise to the denominator.

The second expression is equivalent and can be established by symmetry or similarly by counting how properties of O_i can be chosen given a certain choice of properties of O_j. The third expression in Eq. D.1 is simply an expansion of the combinatorial notation for either of the previous two. ■

The larger result that needs to be derived is, for different values of k, with how many objects in the knowledge base on average will an object share k properties. Clearly, this quantity depends on the number of objects with $1, 2, \ldots, U$ properties (I assume there are no objects with 0 properties in the knowledge base).

Let $E(O_i, k)$ denote the expected number of objects with which a given object O_i will share k properties. $E(O_i, k)$ can be obtained by summing the probabilities

that O_i shares k properties with one object across all objects and subtracting out the object itself. Using the results of Lemma D.1 yields:

$$E(O_i, k) = \left(\sum_{j=0}^{U} N_j \frac{\binom{i}{k}\binom{U-i}{j-k}}{\binom{U}{j}} \right) - \frac{\binom{i}{k}\binom{U-i}{i-k}}{\binom{U}{i}} \tag{D.2}$$

where N_j denotes the number of objects with j properties.

Averaging across all objects yields the expression for the expected number of objects with which a given object O will share k properties:

$$E_{av}(O, k) = \sum_{i=k}^{U} N_i \, E(O_i, k) \Big/ \sum_{i=1}^{U} N_i \tag{D.3}$$

$$= \left(\sum_{i=k}^{U} N_i \left(\left(\sum_{j=k}^{U} N_j \frac{\binom{i}{k}\binom{U-i}{j-k}}{\binom{U}{j}} \right) - \frac{\binom{i}{k}\binom{U-i}{i-k}}{\binom{U}{i}} \right) \right) \Big/ \sum_{i=1}^{U} N_i \tag{D.4}$$

This exact expression may be useful in some situations. In others, it may be more useful to have a simpler approximation. In the remainder of the appendix, I derive approximations to Eqs. D.1 and D.3.

Recall Sterling's approximation for $n!$:

$$n! \approx \sqrt{2\pi n}\, n^n e^{-n}$$

Using this approximation for the factorials involving U, Eq.D.1 can be approximated as follows:

$$\frac{\binom{i}{k}\binom{U-i}{j-k}}{\binom{U}{j}} = \frac{(U-i)!\,(U-j)!\,i!\,j!}{U!\,(U-i-j+k)!\,(i-k)!\,(j-k)!\,k!} \tag{D.5}$$

$$\approx \frac{\sqrt{2\pi(U-i)}\,U^{(U-i)}e^{-U+i}\sqrt{2\pi(U-j)}\,U^{(U-j)}e^{-U+j}}{\sqrt{2\pi(U-j)}\,U^{U}e^{-U}\,\sqrt{2\pi(U-i-j+k)}\,U^{(U-i-j+k)}e^{-U+i+j-k}} \times \frac{i!\,j!}{(i-k)!\,(j-k)!\,k!} \tag{D.6}$$

$$= \sqrt{\frac{(U-i)(U-j)}{U(U-i-j+k)}}\,\frac{U^{(U-i)}\,U^{(U-j)}}{U^{U}\,U^{(U-i-j+k)}e^{-k}}\,\frac{i!\,j!}{(i-k)!\,(j-k)!\,k!} \tag{D.7}$$

$$= \sqrt{\frac{U^2-U(i+j)+ij}{U^2-U(i+j)+Uk}}\,\left(\frac{e}{U}\right)^k\frac{i!\,j!}{(i-k)!\,(j-k)!\,k!} \tag{D.8}$$

$$= \sqrt{\frac{U^2-U(i+j)+ij}{U^2-U(i+j)+Uk}}\,\left(\frac{e}{U}\right)^k k!\binom{i}{k}\binom{j}{k} \tag{D.9}$$

$$\approx \left(\frac{e}{U}\right)^k k!\binom{i}{k}\binom{j}{k} \tag{D.10}$$

$E_{av}(O,k)$ of Eq. D.3 can be approximated by first dropping a minor term and then applying the approximation derived in Eqs. D.5 through D.10 as follows:

$$E_{av}(O,k) \approx \left(\sum_{i=k}^{U} N_i\left(\left(\sum_{j=k}^{U} N_j\frac{\binom{i}{k}\binom{U-i}{j-k}}{\binom{U}{j}}\right) - \frac{\binom{i}{k}\binom{U-i}{i-k}}{\binom{U}{i}}\right)\right) \Bigg/ \sum_{i=1}^{U} N_i \tag{D.11}$$

$$\approx \sum_{i=k}^{U}\sum_{j=k}^{U} N_i N_j\frac{\binom{i}{k}\binom{U-i}{j-k}}{\binom{U}{j}} \Bigg/ \sum_{i=1}^{U} N_i \tag{D.12}$$

$$\approx \left(\sum_{i=k}^{U}\sum_{j=k}^{U} N_i N_j\left(\frac{e}{U}\right)^k k!\binom{i}{k}\binom{j}{k}\right) \Bigg/ \sum_{i=1}^{U} N_i \tag{D.13}$$

$$= \left(\frac{e}{U}\right)^k k!\sum_{i=k}^{U}\left(N_i\binom{i}{k}\sum_{j=k}^{U} N_j\binom{j}{k}\right) \Bigg/ \sum_{i=1}^{U} N_i \tag{D.14}$$

$$= \left(\frac{e}{U}\right)^k k!\left(\sum_{i=k}^{U} N_i\binom{i}{k}\right)\left(\sum_{j=k}^{U} N_j\binom{j}{k}\right) \Bigg/ \sum_{i=1}^{U} N_i \tag{D.15}$$

$$= \left(\frac{e}{U}\right)^k k!\left(\sum_{i=k}^{U} N_i\binom{i}{k}\right)^2 \Bigg/ \sum_{i=1}^{U} N_i \tag{D.16}$$

For k =1, 2, and 3, $E_{av}(O,k)$ can be expanded as follows:

$$E_{av}(O,1) \approx \left(\frac{e}{U}\right)\left(\sum_{i=1}^{U} N_i\, i\right)^2 \Big/ \sum_{i=1}^{U} N_i \tag{D.17}$$

$$E_{av}(O,2) \approx \left(\frac{e}{U}\right)^2 2\left(\sum_{i=2}^{U} N_i \frac{i(i-1)}{2}\right)^2 \Big/ \sum_{i=1}^{U} N_i \tag{D.18}$$

$$= \left(\frac{e}{U}\right)^2 \frac{1}{2}\left(\sum_{i=2}^{U} N_i\, i(i-1)\right)^2 \Big/ \sum_{i=1}^{U} N_i \tag{D.19}$$

$$E_{av}(O,3) \approx \left(\frac{e}{U}\right)^3 6\left(\sum_{i=3}^{U} N_i \frac{i(i-1)(i-2)}{6}\right)^2 \Big/ \sum_{i=1}^{U} N_i \tag{D.20}$$

$$= \left(\frac{e}{U}\right)^3 \frac{1}{6}\left(\sum_{i=3}^{U} N_i\, i(i-1)(i-2)\right)^2 \Big/ \sum_{i=1}^{U} N_i \tag{D.21}$$

These expansions show that as k grows the results of the approximation get increasingly more sensitive to the values N_i for large i. At the same time, Section 7.1 has demonstrated that N_i in the seed knowledge base is well approximated with the power law approximation for $N_i \approx \frac{C}{i^2}$ (for $C = 6789$). This provides a good fit for values of i up to $\sqrt{C} \approx 82$. To approximate values of N_i for $0 < i < U$ in the seed knowledge base and avoid the cumulative influence of fractional number of objects in the approximation for large values of i, I use the piecewise function for N_i:

$$N_i = \begin{cases} \frac{6789}{i^2} & \text{for } 1 \le i \le 82 \\ 0 & \text{for } 82 < i. \end{cases} \tag{D.22}$$

Under this assumption, Eq. D.16 can be rewritten as:

$$E_{av}(O,k) \approx \left(\frac{e}{U}\right)^k k!\, C^2 \left(\sum_{i=k}^{82} \frac{1}{i^2}\binom{i}{k}\right)^2 \Big/ \sum_{i=1}^{U} N_i \tag{D.23}$$

Applying this equation to the specific case of the seed knowledge base amounts to substituting in the values for $U = 32,975, C = 6,789$, and the total number of objects

$\sum_{i=1}^{U} N_i = 12,326$. The approximations for $E_{av}(O,1)$, $E_{av}(O,2)$, and $E_{av}(O,3)$ then evaluate to:

$$
\begin{aligned}
E_{av}(O,1) &\approx \left(\frac{e}{U}\right)\left(\sum_{i=1}^{U} N_i\, i\right)^2 \Big/ \sum_{i=1}^{U} N_i && \text{(D.24)}\\
&\approx \left(\frac{e}{U}\right) C^2 \left(\sum_{i=1}^{82} \frac{1}{i}\right)^2 \Big/ \sum_{i=1}^{U} N_i && \text{(D.25)}\\
&\approx \frac{2.7182}{32975} \times 6789^2 \times 4.99^2 / 12326 && \text{(D.26)}\\
&\approx 7.65 && \text{(D.27)}\\
E_{av}(O,2) &\approx \left(\frac{e}{U}\right)^2 \frac{1}{2}\left(\sum_{i=2}^{U} N_i\, i(i-1)\right)^2 \Big/ \sum_{i=1}^{U} N_i && \text{(D.28)}\\
&\approx \left(\frac{e}{U}\right)^2 \frac{1}{2} C^2 \left(\sum_{i=1}^{82} \frac{i-1}{i}\right)^2 \Big/ \sum_{i=1}^{U} N_i && \text{(D.29)}\\
&\approx \left(\frac{2.7182}{32975}\right)^2 \times \frac{1}{2} \times 6789^2 \times 77.01^2 / 12326 && \text{(D.30)}\\
&\approx 0.07489 && \text{(D.31)}\\
E_{av}(O,3) &\approx \left(\frac{e}{U}\right)^3 \frac{1}{6}\left(\sum_{i=3}^{U} N_i\, i(i-1)(i-2)\right)^2 \Big/ \sum_{i=1}^{U} N_i && \text{(D.32)}\\
&\approx \left(\frac{e}{U}\right)^3 \frac{1}{6} C^2 \left(\sum_{i=1}^{82} i - 3 + \frac{2}{i}\right)^2 \Big/ \sum_{i=1}^{U} N_i && \text{(D.33)}\\
&\approx \left(\frac{2.7182}{32975}\right)^3 \times \frac{1}{6} \times 6789^2 \times 3167^2 / 12326 && \text{(D.34)}\\
&\approx 0.003470 && \text{(D.35)}
\end{aligned}
$$

Given the large magnitudes of the numbers being approximated, the observed results can be said to be in rough agreement with the results obtained by simulation described in Section 7.2. Estimated values $\hat{E}(O,k)$ resulting from averaging ten runs of the simulation, as well as the approximate values derived in this appendix are presented in Table D.1.

k	$E(O,k)$	$\hat{E}(O,k)$
1	7.65	15.58
2	0.07489	0.062
3	0.003470	0.0015

Table D.1: Average number of objects with which an object in the seed knowledge base shares k properties, for $1 \leq k \leq 3$. $E(O,k)$ denotes values obtained from the closed-form approximation, and $\hat{E}(O,k)$ denotes averaged values of ten runs of values obtained by simulation.

Bibliography

Aamodt, A. (1995). Knowledge acquisition and learning from experience—the role of case-specific knowledge. In Tecuci, G. and Kodratoff, Y. (Eds.), *Machine learning and knowledge acquisition; Integrated approaches*, chapter 8, pages 197–245. Academic Press. `ftp://ftp.ifi.ntnu.no/pub/Publikasjoner/vitenskaplige-artikler/kaml-book-95.pdf`. 147

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Proceeedings of the 8th International Conference on Database Theory, ICDT*, volume 1973, pages 420–434. 65

Agrawal, R., Imielinski, T., and Swami, A. N. (1993). Mining association rules between sets of items in large databases. In Buneman, P. and Jajodia, S. (Eds.), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C. 105, 159

Alshawi, H., Carter, D. M., van Eijck, J., Gambäck, B., Moore, R. C., Moran, D. B., Pereira, F. C. N., Pulman, S. G., Rayner, M., and Smith, A. G. (1992). *The Core Language Engine*. Cambridge, Massachusetts: The MIT Press. 62

Atkins, B. T., Kegl, J., and Levin, B. (1986). Explicit and implicit information in dictionaries. In *Proceedings of the Second Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary: Advances in Lexicology*, pages 45–63, Waterloo, Ontario, Canada, N2L 3G1. UW Centre for the New OED, University of Waterloo. 23

Boose, J. H. (1989). A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition*, *1*(1), 3–37. 148

Brachman, R. J., McGuiness, D. L., Patel-Schneider, P. F., and Resnick, L. A. (1990). Living with CLASSIC: when and how to use a KL-ONE-like language. In Sowa, J. (Ed.), *Principles of semantic networks*. San Mateo, US: Morgan Kaufmann. 151

Bruce, R. and Wiebe, J. (1994). Word sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, pages 139–146, LasCruces, NM. 88

Brunk, C. and Pazzani, M. (1991). An investigation of noise-tolerant relational concept learning algorithms. In Birnbaum, L. and Collins, G. (Eds.), *Proceedings of the 8th International Workshop on Machine Learning*, pages 389–393. Morgan Kaufmann. 151

Brunk, C. and Pazzani, M. J. (1992). Knowledge acquisition with a knowledge-intensive machine learning system. 151

Califf, M. E. (1998). Relational learning techniques for natural language extraction. Technical Report AI98-276, UTexas at Austin. 151

Cardie, C., Ng, V., Pierce, D., and Buckley, C. (2000). Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, pages 180–187. Association for Computational Linguistics/Morgan Kaufmann. 152

Chklovski, T. (1998). Recognition and classification by exploration. Master's thesis, Massachussetts Institute of Technology. Available online at `http://citeseer.nj.nec.com/chklovski98recognition.html`. 61

Chklovski, T. and Mihalcea, R. (2002). Building a sense tagged corpus with open mind word expert. In *Proceedings of the Workshop on "Word Sense Disambiguation:*

Recent Successes and Future Directions, pages 116–122, Philadelphia, PA. ACL. Available online at `http://citeseer.nj.nec.com/chklovski02building.html`. 86, 91, 157

Cohen, P. R., Chaudhri, V. K., Pease, A., and Schrag, R. (1999). Does prior knowledge facilitate the development of knowledge-based systems? In *AAAI/IAAI*, pages 221–226. 116

Cohen, P. R., Schrag, R., Jones, E. K., Pease, A., Lin, A., Starr, B., Gunning, D., and Burke, M. (1998). The DARPA high-performance knowledge bases project. *AI Magazine*, *19*(4), 25–49. Available online at `http://citeseer.nj.nec.com/cohen98darpa.html`. 20

Cycorp (1997). Cyc® Ontology Guide: Introduction. Available online at `http://www.cyc.com/cyc-2-1/intro-public.html`. 130

Dagan, I. and Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *International Conference on Machine Learning*, pages 150–157. Available online at `http://citeseer.nj.nec.com/17150.html`. 91

Dahlgren, H., Ljungberg, J., and Ohlund, S. (1991). Access to the repository using natural language knowledge. In *Proceedings of the Second Workshop on Next Generation of Case Tools*, Trondheim. 152

Dahlgren, K., McDowell, J., and Stabler, Jr., E. P. (1989). Knowledge representation for commonsense reasoning with text. *Computational Linguistics*, *15*(3), 149–170. 113, 146, 150

DAML (2002). DARPA Agent Markup Language (DAML). Available online at `http://www.daml.org/about.html`. 151

DARPA (1998). Summary of DARPA workshop on knowledge discovery, data mining, and machine learning (KDD-ML). Available at `http://www.darpa.mil/iso/EELD/KnowledgeDiscovery_DM_andML_Report_for_EELD_(2).doc`. 43

DARPA (2000). Rapid Knowledge Formation (RKF) initiative. Available online at `http://reliant.teknowledge.com/RKF/publication/index.html`. 20

Davis, R. (1979). Interactive transfer of expertise: Acquisition of new inference rules. *Artificial Intelligence, 12*(2), 409–427. 148, 150

Dolan, W., Vanderwende, L., and Richardson, S. D. (1993). Automatically deriving structured knowledge bases from on-line dictionaries. Technical Report MSR-TR-93-07, Microsoft Research, Redmond, WA. Available from `ftp://ftp.research.microsoft.com/pub/tr/tr-93-07.ps`. 23, 24, 128, 152

Dorna, M. and Emele, M. (1996). Semantic-based transfer. In *Proc. 16th Int. Conf. on Computational Linguistics*, pages 316–321, Denmark. ACL. 19

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification* (2nd ed.)., volume November. New York: John Wiley & Sons, Inc. 64, 68, 112

Faure, D. and Nédellec, C. (1998). ASIUM: Learning subcategorization frames and restrictions of selection. In Kodratoff, Y. (Ed.), *10th Conference on Machine Learning (ECML 98) – Workshop on Text Mining*, Chemnitz, Germany. Avril. 152

Federici, S., Montemagni, S., and Pirrell, V. (1996). Analogy-based learning and natural language processing. *ERCIM News*. Available online at `http://www.ercim.org/publication/Ercim_News/enw24/language.html`. 29

Feigenbaum, E. A. (1984). Knowledge engineering: The applied side of artificial intelligence. *Annals of the New York Academy of Sciences, 426*, 91–107. 15

Fellbaum, C. (Ed.). (1998). *WordNet: An Electronic Lexical Database*. Cambridge, Massachusetts: The MIT Press. 18, 34, 130

Fensel, D., Angele, J., and Studer, R. (1998). The knowledge acquisition and representation language KARL. *Knowledge and Data Engineering, 10*(4), 527–550. 151

Forbus, K. D., Falkenhainer, B., and Gentner, D. (1986). The structure-mapping engine. Report, University of Illinois at Urbana-Champaign, Department of Computer Science, Urbana, Illinois. 150

Gaines, B. (1989). An ounce of knowledge is worth a ton of data: quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 156–159, San Mateo, California. Morgan Kaufmann. 116

Gaines, B. and Shaw, M. (1993). Knowledge acquisition tools based on personal construct psychology. *The Knowledge Engineering Review*, *8*(1). 149

Gaines, B. and Shaw, M. (1998). Developing for web integration in sisyphus-iv: Webgrid-ii experience. In Gaines, B. and Musen, M. (Eds.), *Proceedings of Eleventh Knowledge Acquisition Workshop.* 153

Gaines, B. R. (1993). The quantification of knowledge—formal foundations for acquisition methodologies. In Buchanan, B. G. and Wilkins, D. C. (Eds.), *Readings in Knowledge Acquisition and Learning*, section 4.3.1. Morgan Kaufmann. 149

Gaines, B. R. and Shaw, M. L. G. (1992). Integrated knowledge acquisition architectures. *Journal of Intelligent Information Systems*, *1*(1), 9–34. 147

Gentner, D. (1987). Mechanisms of analogical learning. Technical Report 1381, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois. 29

Goldstone, R. (1999). *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*, chapter Similarity, pages 763–765. Cambridge, MA: MIT Press. Available online at `http://cognitrn.psych.indiana.edu/rgoldsto/pdfs/mitecs.pdf`. 64

Guha, R. V. and Lenat, D. B. (1990). Cyc: A midterm report. *AI Magazine*, *11*(3), 32–59. 22

Guha, R. V. and Lenat, D. B. (1994). Enabling agents to work together. *Communications of the ACM*, *37*(7), 127–142. 130

Haase, K. (1996). FramerD: Representing knowledge in the large. *IBM Systems Journal*, *35*(3&4), 381–397. 33

Hahn, U. and Schnattinger, K. (1998). Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531. 152

Han, J. and Fu, Y. (1995). Discovery of multiple-level association rules from large databases. In *Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95)*, pages 420–431, Zürich, Switzerland. 159

Harabagiu, S., Miller, G., and Moldovan, D. (1999). WordNet 2 – a morphologically and semantically enhanced resource. In *Proc. of the SIGLEX Workshop*. Available online at `http://citeseer.nj.nec.com/harabagiu99wordnet.html`. 18

Hearst, M., Hunson, R., and Stork, D. (1999). Building intelligent systems one e-citizen at a time. *IEEE Intelligent Systems [see also IEEE Expert]*, *14*, 16–20. 26, 85, 86, 154

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *In Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING 92)*, pages 539–545, Nantes, France. Also available at `http://citeseer.nj.nec.com/hearst92automatic.html`. 23, 24, 152, 157

Hellerstein, J. M. (1997). Towards a crystal ball for data retrieval. In *Next Generation Information Technologies and Systems*. Available online at `http://citeseer.nj.nec.com/83.html`. 44

Jones, S. (2002). Complete list of spelling rules for nouns and verbs. Available online at `http://www.gsu.edu/~wwwesl/egw/susan.htm`. 174

Jurafsky, D. and Martin, J. (2000). *Speech and Language Processing*. New Jersey: Prentice Hall. 62, 77

Kilgarriff, A. (2002). English lexical sample task description. In *Proceedings of* SENSEVAL-*2, ACL Workshop*. Available online at `http://www.itri.bton.ac.uk/events/senseval/englexsamp.ps`. 90

Kim, J. and Gil, Y. (1999). Deriving expectations to guide knowledge base creation. In *AAAI/IAAI*, pages 235–241. 148

Kim, J. and Gil, Y. (2000). Acquiring problem-solving knowledge from end users: Putting interdependency models to the test. In *AAAI/IAAI*, pages 223–229. Also available at `http://citeseer.nj.nec.com/kim00acquiring.html`. 116

Knight, K. and Luk, S. K. (1994). Building a large-scale knowledge base for machine translation. In *AAAI-94*, pages 773–778, Seattle, WA. AAAI Press. 130

Kwok, C., Etzioni, O., and Weld, D. S. (2001). Scaling question answering to the web. In *WWW-10*. ACM 1-58113-348-0/01/0005. 152

Lamel, L., Gauvain, J., and Adda, G. (2001). Investigating lightly supervised acoustic model training. In *Proceedings ICASSP-01*, Salt Lake City. Available online at `http://citeseer.nj.nec.com/lamel01investigating.html`. 21

Landauer, T. K. (1986). How much do people remember? some estimates of the quantity of learned information in long-term memory. *Cognitive Science*, *10*(4), 477–493. 147

Lenat, D. and Guha, R. V. (1990). *Building Large Knowledge-Based Systems.* Addison-Wesley (Reading MA). 18

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33–38. 18, 19, 22, 145, 150

Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). CYC: towards programs with common sense. *Communications of the ACM, (CACM), August 1990*, *33*(8), 30–49. Available online at `http://www.cyc.com/tech-reports/act-cyc-108-90/act-cyc-108-90.html`. 23, 146

Lenat, D. B., Miller, G. A., and Yokoi, T. (1995). CYC, WordNet, and EDR: Critiques and responses. *Communications of the ACM, 38*(11), 45–48. 18, 19

Lesk, M. E. (1986). Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference.* 89

Li, Q., Shilane, P., Noy, N. F., and Musen, M. A. (2000). Ontology acquisition from on-line knowledge sources. In *AMIA Annual Symposium*, Los Angeles, CA. 153

Lin, D. (1998). An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA. Available online at `http://citeseer.nj.nec.com/95071.html`. 48, 66, 67, 68, 69

Lotka, A. J. (1926). The frequency distribution of scientific productivity. *Journal of the Washington Academy of Sciences, 16*(12), 317–323. 97, 98

Maedche, A. and Staab, S. (2000). Discovering conceptual relations from text. Technical Report 399, Institute AIFB, Karlsruhe University. Available online at `http://citeseer.nj.nec.com/article/maedche00discovering.html`. 152

Maedche, A. and Staab, S. (2001). Ontology learning for the semantic web. *IEEE Intelligent Systems, 16*(2). 152

Mahesh, K. (1996). Ontology development for machine translation: Ideology and methodology. Technical Report MCCS-96-292, Computing Research Lab, New Mexico State University. 130

Marcus, S. and McDermott, J. (1989). SALT: A knowledge acquisition language for propose-and-revise systems. *Artificial Intelligence, 39*(1), 1–37. 148

Menzies, T. (1998). Knowledge maintenance: The state of the art. *The Knowledge Engineering Review, 10*(2). Also avialble at `http://citeseer.nj.nec.com/menzies97knowledge.html`. 43, 44, 148

Mihalcea, R. (2002). Instance based learning with automatic feature selection applied to word sense disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics COLING 2002*, Taiwan. 89

Miller, G. (1998). Nouns in wordNet. In Fellbaum, C. (Ed.), *WordNet: An Electronic Lexical Database*, pages 23–46. Cambridge, Massachusetts: The MIT Press. 55, 130

Minsky, M. (1968). *Semantic Information Processing.* Cambridge, Massachusetts: MIT Press. 146

Minsky, M. (1986). *The Society of Mind.* New York: Simon and Schuster. 27, 44

Minsky, M. (2000). Commonsense-based interfaces. *Communications of the ACM*, *43*(8), 66–73. 23

Moldovan, D. I. and Gîrju, R. (2001). An interactive tool for the rapid development of knowledge bases. *International Journal on Artificial Intelligence Tools*, *10*(1-2), 65–86. Available online at `http://citeseer.nj.nec.com/moldovan01interactive.html`. 19, 24, 25

Montemagni, S. and Pirelli, V. (1998). Augmenting WordNet-like lexical resources with distributional evidence. an application-oriented perspective. In Harabagiu, S. (Ed.), *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 87–93. Somerset, New Jersey: Association for Computational Linguistics. Available online at `http://citeseer.nj.nec.com/65150.html`. 63

Mooney, R. J. (1996). Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In Brill, E. and Church, K. (Eds.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 82–91. Somerset, New Jersey: Association for Computational Linguistics. 89

Morik, K., Wrobel, S., Kietz, J.-U., and Ende, W. (1993). *Knowledge Acquisition*

and Machine Learning - Theory, Methods and Applications (1st ed.). London: Academic Press. 151

Mueller, E. (1999). Prospects for in-depth story understanding by computer. Available at `http://xenia.media.mit.edu/~mueller/papers/storyund.html`. 130

Mueller, E. (2000). ThoughtTreasure: A natural language/commonsense platform. Available at `http://www.signiform.com/tt/htm/overview.htm`. 128, 130

Mueller, E. (2002). OpenCyc vs. ThoughtTreasure: A comparison. Available at `http://www.signiform.com/tt/htm/opencyctt.htm`. 130, 131

Nilsson, N. J. (1995). Eye on the prize. *AI Magazine*, *16*(2), 9–17. Available online at `http://cs.gmu.edu/~zduric/cs580/prize.pdf`. 23

Noy, N. F., Grosso, W., and Musen, M. A. (2000). Knowledge-acquisition interfaces for domain experts: An empirical evaluation of protégé-2000. In *Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE2000)*, Chicago, IL. Also at `http://protege.stanford.edu/papers.html`. 116

OpenCyc (2001). OpenCyc web page. Available at `http://www.opencyc.org/`. 130

Porter, B. and Souther, A. (1999). Knowledge-based information retrieval. In *AAAI Fall Symposium, TR FS-99-02*, pages 81–90. AAAI Press. 22

Qiu, Y. and Frei, H.-P. (1995). Improving the retrieval effectiveness by a similarity thesaurus. Technical Report 225, ETH Zürich, Department of Computer Science, Zürich, Switzerland. Also available at `http://citeseer.nj.nec.com/qiu94improving.html`. 84

Quinlan, J. R. and Cameron-Jones, R. M. (1995). Induction of logic programs: FOIL and related systems. *New Generation Computing*, *13*, 287–312. 151

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453. Also available at `http://citeseer.nj.nec.com/resnik95using.html`. 48, 66, 69

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research, 11*, 95–130. Available online at `http://citeseer.nj.nec.com/resnik99semantic.html`. 48, 69

Resnik, P. and Yarowsky, D. (1997). A perspective on word sense disambiguation methods and their evaluation. In Light, M. (Ed.), *Tagging Text with Lexical Semantics: Why, What and How?*, pages 79–86. Washington: SIGLEX (Lexicon Special Interest Group) of the ACL. 88

Richardson, S., Vanderwende, L., and Dolan, W. (1993). Combining dictionary-based and example-based methods for natural language analysis. Technical Report MSR-TR-93-08, Microsoft Research, Redmond, WA. Available from `ftp://ftp.research.microsoft.com/pub/tr/tr-93-08.doc`. 24, 128, 130, 152

S. G. Pauker, G. A. Gorry, J. P. K. and Schwartz, W. B. (1976). Toward the simulation of clinical cognition: Taking the present illness. *American Journal of Medicine, 60*, 1–18. Available online at `http://medg.lcs.mit.edu/people/psz/HST947_99/PIP-76-5.pdf`. 145

Schwitter, R., Moll'a, D., Fournier, R., and Hess, M. (2000). Answer extraction: Towards better evaluations of nlp systems. In *Proceedings of the Workshop on "Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems", ANLP-NAACL*, pages 20–27, Seattle, Washington. Available online at `http://citeseer.nj.nec.com/295247.html`. 20

Shapiro, S. C. (2000). SNePS: A logic for natural language understanding and commonsense reasoning. In Iwańska, Ł. and Shapiro, S. C. (Eds.), *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowl-*

edge for Language, pages 175–195. Menlo Park, CA: AAAI Press/The MIT Press. 151

Shaw, M. L. G. (1980). *On Becoming A Personal Scientist: Interactive Computer Elicitation of Personal Models Of The World.* London: Academic Press. 149

Singh, P. (2002). The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access.*, Palo Alto, CA. AAAI. Available online at `http://citeseer.nj.nec.com/singh02public.html`. 35, 86, 95, 127, 132, 154

Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., and Zhu, W. L. (2002). Open mind common sense: Knowledge acquisition from the general public. In *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems.* Lecture Notes in Computer Science, Heidelberg: Springer-Verlag. 28, 35, 86, 127

Sleator, D. and Temperley, D. (1993). Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies.* Available through: `ftp://bobo.link.cs.cmu.edu/pub/sleator/link-grammar`. 33, 36, 38

Sommer, E., Morik, K., Andre, J.-M., and Uszynski, M. (1994). What online machine learning can do for knowledge acquisition — A case study. *Knowledge Acquisition, 6*, 435–460. 151

Stork, D. G. and Lam, C. (2000). Open mind animals: Insuring the quality of data openly contributed over the world wide web. In *AAAI Workshop on learning with imbalanced data sets, American Association of Artificial Intelligence Meeting.* Also available at `http://www.openmind.org/OpenMindAAAI.typeset.pdf`. 154

Szolovits, P., Hawkinson, L. B., and Martin, W. A. (1977). An overview of OWL, a language for knowledge representation. Technical Report Tm-86, MIT LCS. 150

Tallis, M., Kim, J., and Gil, Y. (1999). User studies of knowledge acquisition tools: Methodology and lessons learned. In *Proceedings of KAW'99.* 116

Temperley, D., Sleator, D., and Lafferty, J. (2000). Link grammar parser home page. Available at `http://www.link.cs.cmu.edu/link`. 33, 36

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*(4), 327–352. 64

Tversky, A. and Gati, I. (1982). Similarity, separability and the triangle inequality. *Psychological Review, 89*(4), 123–154. 64

Voorhees, E. M. (2000). Overview of the TREC-9 question answering track. In *The Ninth Text REtrieval Conference (TREC 9)*, NIST Special Publication 500-249. NIST. Available at `http://trec.nist.gov/pubs/trec9/papers/qa_overview.pdf`. 83

Watanabe, S. (1969). *Knowing and Guessing - A Formal and Quantitative study.* New York: John Wiley & Sons, Inc. 112

Webb, G. I., Wells, J., and Zheng, Z. (1999). An experimental evaluation of integrating machine learning with knowledge acquisition. *Machine Learning, 35*, 5. 151

Winston, P. H. (1972). Learning structural descriptions from examples. In Winston, P. H. (Ed.), *The Psychology of Computer Vision.* New York: McGraw-Hill. 109, 150

Winston, P. H. (1978). Learning by creating and justifying transfer frames. *Artificial Intelligence, 10*(2), 147–172. 112

Winston, P. H. (1980). Learning and reasoning by analogy. *Communications of the ACM, 23*(12), 689–702. 29

Winston, P. H. (1982). Learning new principles from precedents and exercises. *Artificial Intelligence, 19*(3), 321–350. 150

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Com-*

putational Linguistics (ACL), pages 189–196, Cambridge, Massachusetts. Available online at `http://www.cs.jhu.edu/~yarowsky/acl95.ps`. 89, 90, 91

Ye-Sho Chen, F. F. L. (1986). A relationship between lotka's law, bradford's law, and zipf's law. *Journal of the American Society for Information Science*, *37*(5), 307–314. 97

Yost, G. R. (1993). Knowledge acquisition in Soar. *IEEE Expert*, *8*(3), 26–34. 116

Yuret, D. (1998). *Discovery of linguistic relations using lexical attraction.* PhD thesis, Department of Computer Science and Electrical Engineering, MIT. 84

Zelle, J. M., Mooney, R. J., and Konvisser, J. B. (1994). Combining top-down and bottom-up techniques in inductive logic programming. In Cohen, W. W. and Hirsh, H. (Eds.), *Proceedings of the 11th International Conference on Machine Learning*, pages 343–351. Morgan Kaufmann. 151

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort.* Addison-Wesley (Reading MA). 97, 98

Index