



NORTHWESTERN UNIVERSITY

Electrical Engineering and Computer Science Department

Technical Report
NWU-EECS-07-11
December 31, 2007

Back of the Envelope Reasoning for Robust Quantitative Problem Solving

Praveen Kumar Paritosh

Abstract

Humans routinely answer questions, make decisions, and provide explanations in the face of incomplete knowledge and time constraints. From everyday questions like “What will it cost to take that vacation?” to policy questions like “How can a carbon taxing scheme affect climate change?” we often do not have all the knowledge, time and computational resources to come up with a precise, accurate answer. This thesis describes and formalizes Back of the Envelope (BotE) reasoning – the process of generating rough quantitative estimates.

We claim that a core collection of seven heuristics: mereology, analogy, ontology, density, domain laws, balances and scale-up achieves broad coverage in BotE reasoning. We provide twofold support for this claim: 1) by evaluation of BotE-Solver, an implementation of our theory, on thirty five problems from the Science Olympics, and 2) by a corpus analysis of all the problems on Force and Pressure, Rotation and Mechanics, Heat, and Astronomy from Clifford Swartz's book (2003), “Back-of-the-envelope Physics.”

An aspect of estimation is learning about quantities: what is reasonable, high and low, what are important points on the scale. We call this facility for quantities as *quantity sense*. We present the *Symbolization By Comparison* (SBC) theory of quantity sense. This theory claims that quantity sense consists of qualitative representations of continuous quantity, or symbolizations, which are built by process of comparison. The computational implementation of the SBC theory, CARVE, is evaluated in a functional manner. The representations generated by CARVE help generate more accurate estimates.

NORTHWESTERN UNIVERSITY

Back of the Envelope Reasoning for Robust
Quantitative Problem Solving

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENT

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Praveen Kumar Paritosh

EVANSTON, ILLINOIS

December 2007

UMI Number: 3284176



UMI Microform 3284176

Copyright 2008 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© Copyright by Praveen Kumar Paritosh 2007

All Rights Reserved

ABSTRACT

Back of the Envelope Reasoning for Robust Quantitative Problem Solving

Praveen Kumar Paritosh

Humans routinely answer questions, make decisions, and provide explanations in the face of incomplete knowledge and time constraints. From everyday questions like “What will it cost to take that vacation?” to policy questions like “How can a carbon taxing scheme affect climate change?” we often do not have all the knowledge, time and computational resources to come up with a precise, accurate answer. This thesis describes and formalizes Back of the Envelope (BotE) reasoning – the process of generating rough quantitative estimates.

We claim that a core collection of seven heuristics: mereology, analogy, ontology, density, domain laws, balances and scale-up achieves broad coverage in BotE reasoning. We provide twofold support for this claim: 1) by evaluation of BotE-Solver, an implementation of our theory, on thirty five problems from the Science Olympics, and 2) by a corpus analysis of all the problems on Force and Pressure, Rotation and Mechanics, Heat, and Astronomy from Clifford Swartz's book (2003), “Back-of-the-envelope Physics.”

An aspect of estimation is learning about quantities: what is reasonable, high and low, what are important points on the scale. We call this facility for quantities as *quantity sense*. We present the *Symbolization By Comparison* (SBC) theory of quantity sense. This theory claims that quantity sense consists of qualitative representations of continuous quantity, or symbolizations, which are built by process of comparison. The computational implementation of

the SBC theory, CARVE, is evaluated in a functional manner. The representations generated by CARVE help generate more accurate estimates.

ACKNOWLEDGMENTS

I would like to thank Ken Forbus, my adviser, for his faith in me. For showing by example how to be highly ambitious and pragmatic at the same time. For giving me an incredible opportunity and the freedom to explore and play, both inside and outside the areas of his interest. For sharing his advice and clear perspective.

I would like to thank Dedre Gentner for teaching me cognitive psychology. For listening to me and guiding me despite my ignorance. My fascination with cognition is her doing.

I would like to thank Larry Birnbaum for teaching me semantic analysis. For showing by example how a vast breadth of ideas and associations can be harnessed for solving real problems.

I would like to thank Chris Riesbeck for teaching me how to write beautiful programs. For showing by example how to have simplicity and rigor in ideas and their expression.

I would like to thank Allan Collins for showing by example that science is playing with ideas, writing papers is conversation, and that theory is important.

I would like to thank Tom Hinrichs for taking the time to talk to me as a friend and as an adviser, for letting me in his office, where a significant part of my intellectual and personal experience was deconstructed and ruminated on.

I would like to thank the many other faculty members at Northwestern that I interacted and learned from: Andrew Ortony, Peter Dinda, Vana Doufexi, Ian Horswill, Paul Reber, Ken Paller, Brian Dennis, Jack Tumblin.

I would also like to thank the members of Qualitative Reasoning community, especially Peter Struss, Bert Bredeweg, Paulo Salles, Tim Nuttle, Ben Kuipers and Nuria Agell.

I would like to thank the Artificial Intelligence Program of the Computer Science Division of the Office of Naval Research for financial support, and the Cognitive Science program at the Northwestern University for conference travel support.

Much of the energy, support and excitement came from fellow graduate students and friends. I would like to thank Jason Skicewicz, Mike Knop, Scott Hoover, Rob Zubek, Sven Kuehne, Pinku Surana, Ke Cheng, Robin Hunicke, Ron Ferguson, Tom Bittner, Mike Brokowski, Karen Carney, Leo Ureel, Jay Budzik, Patricia Dyck, Matt Klenk, Andrew Lovett, Dan Halstead, Morteza Dehgani, Kate Lockwood, Emmett Tomai, Jeff Usher, Manan Sanghi, Earl Wagner, Kevin Livingston, Shaji Chempath, Sumit Kewalramani, Ranjith Nair, Reuben Thomas, Mohit Singh and Anand Priyadarshee. I cannot thank them enough for sharing the highs, the lows and the nothingnesses.

I would like to thank Julie Saltzman for all her support, for listening to me, reading my prose, and being there. Without her, these years would have been much less fun.

I would like to thank my father for dreaming and my mother for figuring out how to make it true. And my sisters, Lilli, Sweetie, Guria, Soni and Lado for playing with me, letting me teach them, and most importantly, annoy them.

TABLE OF CONTENTS

Chapter 1: Introduction.....	16
1.1 Brittleness in Artificial Intelligence.....	17
1.2 Heuristic Reasoning for Alleviating Brittleness.....	18
1.3 Back of the Envelope Reasoning.....	20
1.4 A Computational Model of BotE Reasoning.....	21
1.5 Claims and Contributions.....	23
1.6 Roadmap.....	24
Chapter 2: A Theory of Back of the Envelope Reasoning.....	25
2.1 Introduction	25
2.2 Examples of Back of the Envelope Reasoning.....	25
2.3 Motivation.....	29
2.4 Qualitative Reasoning and Commonsense Reasoning.....	31
2.5 A Model of BotE Reasoning.....	34
2.5.1 Direct Estimation	35
2.5.2 Estimation Modeling	36
2.6 Representation of BotE Problems.....	37
2.7 Representation of Heuristic Methods.....	38
2.7.1 Object-based Heuristic Methods.....	40
2.7.1.1 Mereology.....	40
2.7.1.2 Similarity.....	41

2.7.1.3 Ontology.....	43
2.7.2 Quantity-based Heuristic Methods.....	44
2.7.2.1 Density.....	44
2.7.2.2 Domain Laws	44
2.7.3 System-based Heuristic Methods.....	45
2.7.3.1 System Laws	46
2.7.3.2 Scale-up	46
2.7.4 Reasonableness and Comprehensiveness.....	46
2.8 Corpus Analysis of Swartz's Back-of-the-Envelope Physics.....	48
2.9 Summary.....	51
Chapter 3: A Theory of Quantity Sense.....	52
3.1 Introduction.....	52
3.2 Quantity Sense.....	54
3.2.1 Definitions and Terminology.....	54
3.2.2 The Space of Quantitative Knowledge.....	56
3.3 Background and Motivation.....	59
3.3.1 Education	60
3.3.2 Linguistics.....	61
3.3.3 Qualitative Representations of Quantity.....	63
3.3.4 Relevant Psychological Phenomena.....	64
3.3.4.1 Context sensitivity.....	64
3.3.4.2 Reference Points and Categorization Effects in Comparison.....	65

3.3.4.3 Models of Retrieval, Similarity and Generalization.....	65
3.4 Representing Quantity Sense.....	67
3.4.1 Constraints.....	68
3.4.1.1 Reasoning Constraints.....	68
3.4.1.2 Ecological Constraints	70
3.4.2 Proposed Representation	72
3.4.2.1 Symbolic references to quantity.....	72
3.5 Necessary, relevant, and more distinctions.....	73
3.5.1 Structural Limit Points.....	74
3.5.2 Distributional Partitions.....	76
3.5.3 Implications	77
3.6 CARVE: Symbolization by Comparison.....	78
3.6.1 Identifying Quantities and Scales.....	80
3.6.2 Distributional Partitioning.....	82
3.6.3 Structural Partitioning and Projection.....	83
3.7 Analogical Estimation.....	84
3.7.1 Verbal Protocols of Analogical Estimation.....	87
3.7.2 A Theory of Analogical Estimation	89
3.7.2.1 Analogical Anchors.....	90
3.7.2.2 Causal Adjustments.....	91
3.7.2.3 Adjustment based upon non-alignable features.....	93
3.7.3 KNACK: A Computational Model of Analogical Estimation.....	93

3.7.4 Estimating Basketball Statistics.....	94
3.8 Conclusions.....	97
Chapter 4: System Description and Examples.....	99
4.1 BotE-Solver.....	100
4.1.1 Knowledge and Reasoning Infrastructure.....	102
4.1.2 Implementing Heuristic Methods Via Suggestions.....	103
4.1.3 Tracking problem solving progress by AND/OR trees.....	105
4.2 Evaluating BotE-Solver.....	109
4.3 Heuristic Methods in Bote-Solver.....	112
4.3.1 Ontology Heuristic Method.....	112
4.3.2 Mereological Estimation Heuristic Method.....	115
4.3.3 Analogy Heuristic Method.....	117
4.3. 4 Density Heuristic Method.....	118
4.3.5 Domain Laws Heuristic Methods.....	119
4.3.6 Scale-Up Heuristic Method.....	120
4.3.7 System Laws Heuristic Method.....	121
4.4 BotE-Solver at the Science Olympics.....	122
4.5 Discussion and Conclusions.....	126
Chapter 5: Related Work.....	129
5.1 Heuristics	129
5.2 Automated Problem Solving.....	131
5.2.1 SAINT	131

5.2.2 FERMİ	132
5.2.3 TPS.....	133
5.2.4 Semi-quantitative Reasoning.....	134
5.3 Psychology of Human Reasoning.....	134
5.3.1 Plausible Reasoning.....	134
5.3.2 Heuristics and Biases	135
5.3.3 Simple Heuristics that Make Us Smart.....	136
5.3.4 Education.....	137
Chapter 6: Conclusions and Future Work.....	139
6.1 Summary of Key Contributions	140
6.1.1 A Broad Coverage Theory of Back of the Envelope (BotE) Reasoning	140
6.1.2 A Cognitively Plausible Theory of Learning about Quantities	141
6.1.3 A Theory of Analogical Estimation	142
6.1.4 Summary	143
6.2 Future Work.....	144
6.2.1 Applications.....	144
6.2.2 Natural Language and Question Answering.....	144
6.2.3 Heuristic Reasoning.....	146
6.2.3.1 Heuristic Domains.....	147
6.2.3.2 Summary.....	152
6.2.4 Cognitive and Educational Implications.....	153
6.3 Final Words.....	154

References.....	155
Appendix A: Suggestions used by BotE-Solver.....	166
Appendix B: Sample Case from Basketball Domain.....	176

TABLE OF FIGURES

Figure 2.1: A simplified schematic of knowledge based system.....	30
Figure 2.2: Quantity Sense and Heuristic Reasoning in BotE.....	35
Figure 3.1: A schematic of a subset of space of research on quantity in Cognitive Science.....	57
Figure 3.2: A schematic of CARVE.....	80
Figure 3.3: Finding structural limit points by projection.....	84
Figure 3.4: The KNACK algorithm.....	94
Figure 3.5: Comparison between normalized mean error of estimates by dimension.....	96
Figure 3.6: Comparing under adjustments, over adjustments, and wrong adjustments made by Knack over all the estimation problems.....	97
Figure 4.1: Some examples from the Science Olympics corpus.....	100
Figure 4.2: Architecture of BotE-Solver.....	101
Figure 4.3. An example suggestion.....	103
Figure 4.4: Predicate calculus statements generated by the SphericalVolume suggestion.....	105
Figure 4.5: The ontology heuristic in BotE-Solver.....	114
Figure 4.6 The HomogenousGroupExtensiveQuantityStrategy suggestion.....	116
Figure 4.7: The CountViaConstituentStrategy suggestion.....	117
Figure 4.8: The DensityStrategy suggestion.....	119
Figure 4.9: The TotalEnergyConversionStrategy suggestion.....	120
Figure 4.10: The SeekScaleModelStrategy suggestion.....	120
Figure 4.11: The AllometricScalingLaw suggestion.....	121
Figure 4.12: The BalanceStrategy suggestion.....	122

TABLE OF TABLES

Table 2.1: Distribution of heuristic methods in Swartz's book.....	50
Table 3.1: Number of analogical estimation occurrences.....	88
Table 4.1: The Science Olympics questions and BotE-Solver's answers.....	122

On two occasions I have been asked, "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.

- Charles Babbage (1791-1871)

Chapter 1: Introduction

Humans routinely answer questions, make decisions, and provide explanations in the face of incomplete knowledge and time constraints. From everyday questions like “What will it cost to take that vacation?” to policy questions like “How can a carbon taxing scheme affect climate change?” we often do not have all the knowledge, time and computational resources to come up with a precise, accurate answer. In case of the vacation example, currency exchange rates may fluctuate, you might not yet know exactly what you will do, and so on. In the carbon tax example, constructing a detailed model of the economics and physics of human climate interaction is a Herculean task – see for example, the seven hundred pages long *Stern Review* on the economics of climate change [Stern 2007]. Even if all the knowledge and computational resources needed to pinpoint the answer were there, in many situations, there is considerable value in a quick but reasonable estimate while one waits for the perfect answer. Examples include time-constrained situations involving taking an action and sanity-checking noisy data, for example, knowledge acquired from reading from the web.

1.1 Brittleness in Artificial Intelligence

A key aspect of human intelligence is the ability to operate flexibly and robustly by making educated guesses and plausible explanations, when perfect correct reasoning cannot be accomplished [Collins and Michalski, 1989]. On the other hand, most Artificial Intelligence (AI) systems, and software in general suffer from brittleness. This brittleness manifests as: 1) they fail to respond at all, because of lack of knowledge or computational resources, or, 2) they generate unreasonable answers, because of inaccuracies in their knowledge. The unfortunate aphorism of “Garbage In, Garbage Out” highlights the fact that computer programs will unquestioningly process their input. One reason why humans, for most part, are exempt from that dictum is because of our ability to make educated guesses to see if something *makes sense*.

Doug Lenat [1986] called this the *brittleness bottleneck*, and proposed that the solution was a large explicit commonsense knowledge base. The current state of this effort is the Cyc knowledge base, which consists of over three million assertions represented in predicate calculus. However, both knowledge gaps and inferential complexity of reasoning are still crippling factors. Project Halo was an effort to systematically analyze the capabilities and limitations of knowledge-based systems. Three different teams: Cycorp, SRI and Ontoprise were given six months to design a system for answering a subset of AP level chemistry questions [Friedland and Allen, 2004]. For 49% of the questions, the Cycorp team failed to come up with an answer, and at times came up with sixteen pages of justifications without a successful answer.

A very different approach that attempts to short-circuit the knowledge issue is to directly tap into vast text corpora: web pages, newspaper articles and scientific papers, to name a few. However, this text-based approach has very little if any capability to produce explanations,

sanity-check answers and make inferences. For instance, in an evaluation of question-answering programs that mine text for answers, one program came up with 360 tons as the amount of Folic acid that an expectant mother should have per day¹, and 14 feet as the diameter of the earth!

Continued progress in both the knowledge-based and text-based approach should facilitate systems that can perform in a broad range of domains. However a source of flexibility in human reasoning is the ability to make educated guesses when knowledge fails. At the risk of being approximate, this extends the usefulness of existing knowledge.

1.2 Heuristic Reasoning for Alleviating Brittleness

This thesis proposes the *heuristic reasoning* approach [Paritosh, 2006] for fixing brittleness: endow the systems with heuristic methods so that it can make educated guesses when other mechanisms fail. Heuristic methods exploit the information processing structure of the reasoning system and the structure of the environment. In problem solving, heuristic methods generate approximate answers in two ways: 1) by using a subset of information needed, and 2) by using a proxy for the information needed. For example, consider the problem of estimating density of the human body. To use a subset of information to infer the density would be to say that since the largest component of the human body is water, its density should be close to that of water, 1000 kg.m⁻³. To use a proxy would be to say that human body density should be close to that of water as they can float/drown in water, and therefore finding the density of water gives us an answer to the original problem.

The notion of heuristics has enjoyed a long history in AI in various guises. Before AI, George Polya [1945] popularized heuristics in his book as steps one could take while solving

¹ The question is from TREC9, and this was reported in the IBM TJ Watson AQUAINT Briefing.

mathematical problems. Some of his heuristics included drawing a figure, working backward from what is to be proved and considering a more general version of the problem. Newell later wrote about the influence or lack thereof of Polya's work on AI [Newell, 1981]. Herb Simon invented the notion of *Bounded Rationality* and *Satisficing* [Simon, 1957]. In this approach, reasoning is still governed by laws of rationality and realistic resource constraints are placed on it. Newell and Simon [1963] proposed weak methods, e.g., means-end analysis, generate and test, etc., as the basis of intelligence. Doug Lenat's AM and Eurisko [1982] systems made scientific discoveries in the domain of mathematics, device physics, games, and heuristics itself, among others, armed with a library of hundreds of heuristics. Lenat called for a formal study of the science of heuristics, heuristics. Lenat's systems used heuristics to make interesting scientific conjectures and in guiding exploration of the large search space. By heuristic reasoning, we mean the inference patterns that support educated guessing. Our context is problem solving, and we want to quickly find a reasonable answer, even if all the relevant knowledge is not available.

A *heuristic domain* is a reasoning task that is amenable to heuristic reasoning. Reasoning tasks where there are multiple answers and/or processes to arrive at the answer, with varying degrees of correctness or quality are heuristic domains. In contrast, questions like "What two US biochemists won the Nobel prize in 1992?" or "What is the scientific name of Viagra?" are examples for which it is less likely to have reasonable guesses – you either know the answer or don't. A *heuristic method* is a specific pattern of reasoning that yields a reasonable inference in its heuristic domain.

The heuristic domain that is described in this thesis is *Back of the Envelope* (BotE) reasoning. BotE reasoning involves the estimation of rough but quantitative answers to questions

where the models and the data might be incomplete. The name back-of-the-envelope comes from the idea of doing quick calculations on the nearest available piece of paper. BotE reasoning exemplifies heuristic reasoning, as it is often invoked in situations where accurate answers are expensive to obtain, and carefully done, without such an expense it can be used to provide high-quality estimates.

To show the feasibility of the heuristic reasoning approach, we have built *BotE-Solver*, a system that solves estimation questions like “How much money is spent on healthcare in the US?” BotE-Solver can currently solve problems from both commonsense and scientific domains, and we evaluate it on problems from the Science Olympics, a competition for high school students (Division “C” in the US).

The next section describes the BotE domain. Section 1.3 presents our approach to model BotE reasoning. Section 1.4 presents the claims and contributions of this work. We conclude with a roadmap for the rest of this thesis.

1.3 Back of the Envelope Reasoning

A rough estimate generated quickly is more valuable and useful than a detailed analysis, which might be unnecessary, impractical, or impossible because the situation does not provide enough time, information, or other resources to perform one. BotE reasoning is useful and practical. In domains like engineering, design, or experimental science, one often comes across situations where a rough answer generated quickly is more valuable than waiting for more information or resources. Some domains like environmental science [Harte, 1988] and biophysics [O’Connor and Spotila, 1992] are so complex that BotE analysis is often the best that can be done with the

available knowledge and data. Additionally, this type of reasoning is particularly common in engineering practice and experimental sciences, including activities like evaluating the feasibility of an idea, planning experiments, sizing components, and setting up and double-checking detailed analyses.

BotE reasoning is ubiquitous in everyday problem solving as well. Common sense reasoning often hinges upon the ability to rapidly make approximate estimates that are fine-grained enough for the task at hand. We live in a world of quantitative dimensions, and reasonably accurate estimation of quantitative values is necessary for understanding and interacting with the world. How long will it take to get there? Do I have enough money with me? These everyday, common sense estimates utilize our ability to draw a quantitative sense of world from our experiences. We believe that the same processes underlie both these commonsense estimates and expert's BotE reasoning to generate rough estimates. Fundamental to both types of reasoning is the process of drawing upon experience to make such estimates, and the achievement of expertise in part by accumulating, organizing, and abstracting from experience to provide the background for such estimates.

1.4 A Computational Model of BotE Reasoning

Our model of BotE reasoning consists of two distinct processes. First, *estimation modeling* is the process of constructing simplified models of complex scenarios which are good enough for the purposes of making a rough estimate. In our theory, these models are constructed by applying heuristic methods. For instance, one of the heuristic methods is *Mereology*, which exploits the

part-whole structure. Given a question like “What is the population of Planet Earth?” the mereology heuristic suggests looking at all the sub-parts, in this case, countries, finding the populations of individual countries and adding them up. If the quantity in question is an extensive quantity like count or mass, adding up is the right way to combine the values; if it is an intensive quantity, like density, the weighted average is the right combinator. It also requires making a closed world assumption, namely that we know all the sub-parts; and that there is no overlap between parts. There are seven heuristic methods (including mereology) used by BotE-Solver. These heuristics are described in Chapter 2, and the implementation details are in Chapter 4.

Second, *direct estimation* is the process of coming up with a numeric value for a quantity. The estimation modeling step is only building a model: to get to an estimate, the process has to bottom out by plugging in numeric values. The simplest case for direct estimation is when the value is explicitly available. But when this is not the case, good human estimators are able to use their knowledge of similar situations to infer a reasonable value for the quantity in question. For instance, to estimate the height of Jason Kidd, the basketball player, one might use the fact that basketball players are usually tall. An even better estimate might be constructed by using the more specific fact that Jason Kidd is a point guard, and basing it on other players similar to Jason Kidd, for instance Steve Nash. We call this facility for quantities built out of experience *quantity sense*. In Chapter 3, we present the *symbolization by comparison* theory of quantity sense, a cognitively plausible account that claims that quantity sense consists of a symbolization of the continuous quantity built by processes of comparison. This has been implemented in a

computational model, CARVE. The representations generated by CARVE lead to more accurate analogical estimates.

1.5 Claims and Contributions

The key theoretical claims of this thesis are:

1. *There is a core collection of powerful heuristic methods that provide broad-coverage in BotE reasoning [Paritosh and Forbus, 2005].* Specifically, the seven heuristic methods of mereology, analogy, ontology, density, domain laws, balances and scale-up, achieve broad coverage in answering BotE questions. This is a *knowledge level* analysis of BotE reasoning [Newell, 1982]. We provide two-fold support for this claim: 1) by evaluation of BotE-Solver’s performance on problems from Science Olympics, and 2) a corpus analysis of all the problems on Force and Pressure, Rotation and Mechanics, Heat, and Astronomy from Clifford Swartz's book (2003), “Back-of-the-envelope Physics.”
2. *Quantity sense consists of qualitative representations of continuous quantity, or symbolizations, which are learned by process of comparison [Paritosh, 2003, 2004].* We call this the symbolization by comparison theory. In Chapter 3, we describe ecological, psychological and task constraints on quantity sense. We provide support for this claim by improved performance on the analogical estimation task owing to the qualitative representations of quantity.

The following systems were built as a part of this thesis:

1. BotE-Solver: A problem solver that uses heuristic methods and the Cyc knowledge base to solve BotE problems [Paritosh and Forbus, 2004, 2005, *in preparation*].
2. CARVE : A system that learns symbolic representations of quantities by exposure to examples in that domain [Paritosh, 2003, 2004].
3. KNACK: A system that implements analogical estimation, i.e., generating numeric estimates for an unknown parameter by finding other similar example(s) [Paritosh and Klenk, 2006].

1.6 Roadmap

Chapter 2 begins by describing BotE reasoning and the importance of this domain. It then presents a formalization of problems and heuristic methods and presents the seven heuristic methods. It then presents the corpus analysis of problems from Swartz's book. Chapter 3 presents a theory of quantity sense. It begins by describing quantity sense and the space of quantitative knowledge organized by a review of relevant literature from education, linguistics, psychology and qualitative reasoning. Based on these, it then proposes constraints on human representations of quantity. It then describes CARVE, the computational model for learning these representations from examples. Chapter 4 describes the BotE-Solver system. It describes implementation aspects of each of the heuristic methods, and the results of evaluation on the Science Olympics corpus of problems. Chapter 5 describes related work. Chapter 6 presents conclusions and future directions.

Chapter 2: A Theory of Back of the Envelope Reasoning

2.1 Introduction

Back of the envelope (BotE) reasoning involves generating quantitative answers in situations where exact data and models are unavailable and where available data is often incomplete and/or inconsistent. This chapter presents a knowledge level [Newell 1982] theory of BotE reasoning. We begin with two extended examples of BotE reasoning in the next section. Then we discuss motivations and broader implications of BotE reasoning. We then compare BotE reasoning to qualitative reasoning, highlighting the commonalities and the differences. Next we present a computational theory of BotE reasoning based on an analysis of different kinds of knowledge and reasoning involved. We present a formalization of BotE problems and heuristic methods, and then we describe seven heuristic methods that we claim achieve broad coverage in answering BotE questions. We present supporting evidence for this claim by a corpus analysis of all the problems (n=44) on Force and Pressure, Rotation and Mechanics, Heat, and Astronomy from Clifford Swartz's book (2003), "Back-of-the-envelope Physics."

2.2 Examples of Back of the Envelope Reasoning

Consider the following examples of BotE questions:

1. How many K-8 school teachers are there in the US?
2. What is the annual cost of health care in the US?
3. What is Jason Kidd's point per game for this season?
4. How much is spent on newspapers in the US per year?

What these questions have in common is: 1) they seek numeric answers, and 2) even though exact answers might be hard to find, it is possible to generate good enough rough estimates. Let's start with a much simpler example. Consider the question: How many pieces of popcorn would fill the room you are now sitting in? You probably don't have a value for the number of popcorns in your memory. One way to estimate it would be –

$$\text{number of popcorns} = \text{volume of room} / \text{volume of popcorn kernel} \quad (1)$$

Approximating room to a cuboid, and a popcorn kernel to a cube (considering the voids left after packing in popcorn kernels this is a reasonable assumption),

$$\text{number of popcorns} = l * b * h / a^3 \quad (2)$$

where l , b , h are length, breadth and height of the room and a is the edge of the cube that describes a popcorn. In (2), we have built an *estimation model* for the number of popcorn kernels, which we have now described in terms of a set of parameters that can be directly estimated. Estimation modeling is a recursive process that continues until we bottom out with parameters that can be successfully directly estimated. At this point, if we do know the values for l , b , h and a above, we can plug those in and get an answer. What if we don't have those values? Let's look at h , the height of the room. One strategy would be to say that it is around twice the

height of a person, so roughly 10 feet high. Or, one could use previous knowledge to say that this room is quite similar to their apartment with respect to height, which they know is 10 feet high.

Let's work out one of the questions given earlier: How many K-8 school teachers are there in USA? If we don't know the number of K-8 teachers, then we need to relate it to other quantities that we do know. One approach is the following –

Number of teachers = number of students / students per teacher

Number of students = population * fraction in the age range of K-8 students *
fraction of kids who go to school

Fraction of population in the K-8 age range = K-8 age range / life expectancy

The above estimation model relates the number of teachers to other known quantities. Estimation models often make simplifying assumptions, e.g., the calculation of the fraction of students assumes a uniform distribution of population with respect to age. In order for the estimation model to successfully produce an answer for the original question, it has to bottom out with numeric values. The next step involves *direct estimation* of the quantities in the estimation model.

Population = 300 million

Life expectancy = 75

Fraction of kids going to school = 1

Students per teacher = 25

The goal of the direct estimation step is to find a numeric value for the given quantity, which is good enough for the estimation task at hand. Some values like the population of the US might be explicitly known, whereas some values like students per teacher might be inferred based on one's classroom experiences. The simplest form of direct estimation is lookup, however it can involve extrapolating from one or more similar examples or categorical information. For example, one

might come up with an estimate for life expectancy by considering some specific examples. At this point, we need to plug in these numbers into the estimation model to obtain the final answer.

$$\text{Fraction of population in the K-8 age range} = \text{K-8 age range} / \text{life expectancy} = 9/75$$

$$\text{Number of students} = \text{population} * \text{fraction in the age range of K-8 students} *$$

$$\text{fraction of kids who go to school}$$

$$= 300 \text{ million} * 9/75 * 1$$

$$\text{Number of teachers} = \text{number of students} / \text{students per teacher}$$

$$= 40 \text{ million} / 25 = 1.6 \text{ million}$$

This last step is often simple arithmetic and/or algebra. For example, in estimating the area of a letter sized sheet of paper (8.5x11), it might be easier to approximate it as the simpler multiplication, 9x10. As arithmetic is the easiest step for a computer program, and we will not spend time in this thesis about discussing or modeling this aspect of human estimation process, (see LeFevre, Greenham and Waheed (1993) for research on this topic). An explanatory simplification above was that the entire estimation modeling was done separate from the direct estimation – as we will see later, these two processes are interleaved.

In this example, the correct answer is 1.9 million according to the Statistical Abstracts of the United States, 2003. Our answer of 1.6 million is off by fifteen percent, which brings us to the issue of correctness in BotE reasoning. Qualitatively, the resolution in BotE reasoning is higher than order-of-magnitude reasoning, and the expected answer is within a factor of two. Most published examples of BotE reasoning demonstrate within factor of two resolution in answer: e.g., regularly appearing columns in American Journal of Physics [Hobart 1963; Purcell 1983-85; Weiskopf 184-86] and Journal of Geological Education [Triplehorn 1994-95], among others.

2.3 Motivation

Brittleness is a serious problem for most AI programs, and software in general. The two common manifestations of brittleness are: 1) the software cannot find an answer, because of gaps in the knowledge base, or because of a lack of required computational resources; and 2) the software comes up with an unreasonable answer, possibly because of inaccuracies in its knowledge base. For instance, in an evaluation of question-answering programs that mine text for answers, one program came up with 360 tons as the amount of Folic acid that an expectant mother should have per day, and 14 feet as the diameter of the earth!²

Knowledge, specifically commonsense knowledge, was proposed as a solution to avoid just these types of failures [Lenat et al., 1986]. One premise of the Cyc project is that by explicitly representing commonsense knowledge, one can build more flexible systems, where commonsense fills in the gaps when the system comes to a point where it would otherwise exhibit brittle behavior. Knowledge based systems consist of reasoning mechanisms that use an explicit *knowledge base*, a database of facts, to answer queries. However, these arguments might apply even more broadly. Figure 2.1 shows a highly simplified view of a knowledge based system. The reasoning mechanisms might consist of forward and backward chaining, planning, analogy, spatial reasoning, and special-purpose procedural attachments to handle specific tasks. Many of these reasoning methods are computationally complex, and in theory can take unbounded amounts of time. However, a crucial bottleneck for these reasoning mechanisms is the knowledge base. If the knowledge base has gaps, i.e., lacks relevant knowledge, then there is no hope of being able to find an answer.

² The questions are from TREC9, and were reported in the IBM TJ Watson AQUAINT Briefing.

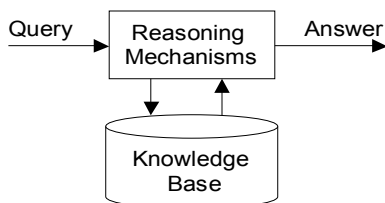


Figure 2.1: A simplified schematic of knowledge based systems

Cyc, the largest knowledge representation effort, consists of over 3 million assertions represented in predicate calculus. Openmind Commonsense³, another such effort, consists of 800,000 assertions in English authored by volunteers on the web (Singh et al., 2002). The innovative idea in this project is that by lowering the barrier to knowledge authoring, it might be possible to quickly build a large collection of commonsense knowledge. However, the problem of inaccuracies in the knowledge base is a serious problem with about 30% of the knowledge being “garbage” (Lieberman, personal communication). Furthermore, it supports very weak notions of reasoning with these facts, if any at all.

Project Halo⁴ is an effort to systematically analyze the capabilities and limitations of knowledge-based systems. Three different teams: Cycorp, SRI and Ontoprise were given six months to design a system for answering a subset of AP level chemistry questions. The detailed results of this evaluation are summarized in Friedland and Allen⁵ (2004). For 49% of the questions, the Cycorp team failed to come up with an answer, and at times coming up with sixteen pages of justifications without a successful answer. A broad commonsense knowledge base is necessary for building robust programs. However, commonsense might be much vaster than imagined, and manually building large databases of knowledge is not enough by itself.

³<http://openmind.media.mit.edu/>

⁴ <http://www.projecthalo.com>

⁵ http://www.projecthalo.com/content/docs/halopilot_vulcan_finalreport.pdf

Humans face the same sources of brittleness: knowledge gaps and inferential complexity. They cope via their remarkable ability to generate educated guesses, reasonable explanations and ballpark estimates when we run into situations where knowledge and/or cognitive resources are lacking. BotE reasoning is an instance of such flexible reasoning. We present a computational theory of BotE reasoning that operationalizes patterns of heuristic reasoning that can flexibly handle gaps in the knowledge base at the cost of being right most, although not all, of the time. This is a different and complementary approach to alleviating brittleness: build heuristic reasoning mechanisms that scale with respect to relevant knowledge and computational resources available.

2.4 Qualitative Reasoning and Commonsense Reasoning

One of the original motivations of qualitative reasoning (QR) was to understand and model human commonsense reasoning [de Kleer & Brown 1984; Forbus 1984; Bredeweg & Schut 1991, White & Frederiksen 1990]. QR helps us determine what phenomena are relevant, and experiential knowledge supplies useful default and pre-computed information, including both numeric values and relevant modeling assumptions, as well as knowledge about similar situations that can serve as a reality check for the estimates. The need to compare parameters and to make estimates guided by similarity in turn raises interesting questions about what role(s) quantitative dimensions play in our judgments of similarity, and how we develop our quantitative sense of a domain with experience. To reiterate, qualitative representations are essential for BotE reasoning for two reasons:

3. *Qualitative models provide analytic framework:* Understanding what entities and physical processes are relevant is crucial in determining what parameters are relevant. Modeling assumptions expressed in terms of the conceptual understanding of the situation determine when particular quantitative estimation techniques are appropriate.
4. *Qualitative models facilitate comparison:* Similarity in qualitative, causal structure helps determine what experience is relevant when making an estimate. Similarity is also used in helping evaluate the reasonableness of an estimate. Including qualitative descriptions in remembered experiences along with quantitative data facilitates comparison and abstraction from experiences.

However, some of the central assumptions of QR in practice must be rethought when considering commonsense reasoning, as opposed to narrow domain expertise. It is commonplace in QR to assume that a domain theory is complete. This assumption is implausible for commonsense reasoning, whether or not one views QR purely in terms of a component in a performance system or as a psychological model. The closer one looks at human knowledge, the more it appears that it is fragmentary, and more concrete than abstract [Forbus and Gentner, 1997]. It may be that such an organization is a necessity for human-level performance, whether or not one is making psychological claims. Let us call this approach Commonsense QR (CQR) for concreteness. Here are the five important constraints shared by CQR and BotE reasoning, but are violated by most traditional QR approaches:

1. *Incompleteness:* Domain theories are incomplete in terms of their coverage, and even what they do cover might contain inconsistencies. Humans have an amazing ability to

make educated guesses even when their knowledge contains gaps. At the cost of being reasonable (instead of accurate) humans can efficiently and robustly reason with fragmentary and incomplete knowledge.

2. *Concreteness*: Domain knowledge includes knowledge of many concrete, specific situations. These concrete descriptions are used directly in analogical reasoning, in addition to first-principles reasoning.
3. *Experiential improvement*: Domain expertise improves through the accumulation of information, both concrete and abstract. Experience improves our abilities to reason through similar situations, and helps us develop intuitions for what is reasonable.
4. *Focused reasoning*: Instead of maintaining uncertainty and ambiguity for completeness, assumptions are made aggressively to tightly constrict the number of possibilities considered. Common sense reasoning is required for action in the world, and there are opportunities for interaction and further reflection, reducing the amount of stress on any particular computation. In most situations, it is better to answer rapidly and sometimes be wrong, than to answer slowly and vaguely.
5. *Pervasively quantitative*: Our interaction with the real world requires concrete choices for quantities. For example, the amount of salt one adds while cooking a certain dish (or an estimate of the increase in Earth's temperature due to greenhouse gases) cannot be safely specified as “+”. While there are certainly tolerances, and we believe that estimation requires drawing upon lots of examples, our actions in the end require that estimates manifest as exact values. Quite possibly this is true for every step along the way, as per the focused reasoning constraint.

Forbus and Gentner (1997) proposed a similarity-based hybrid model of qualitative simulation where analogical reasoning and first-principles qualitative reasoning are tightly interwoven, which has been further explored by Yan and Forbus (2004). Consider predicting what might happen when one is filling a cup with coffee: if there is a positive inflow when the level of coffee reaches the height of the cup, it will overflow. The key insight in the hybrid qualitative simulation proposal is that one can make the same prediction using a previous experience of overfilling the cup rather than deriving it via first principles. This approach addresses the first three constraints above. This thesis focuses on the last two.

2.5 A Model of BotE Reasoning

We present a model of BotE reasoning that captures the two types of knowledge and reasoning involved in it. First, BotE reasoning requires quantitative knowledge and ability to reason with quantities. Without any knowledge of quantitative facts, it will be impossible to answer any BotE question. Direct estimation uses knowledge about quantities and values. With expertise in a domain, one accumulates more and more quantitative facts, and the ability to generalize and extrapolate from them. We call this combination of quantitative knowledge and quantitative reasoning abilities *quantity sense*. Second, BotE reasoning requires knowledge of simplifying heuristics and the ability to reason with them. We call this *heuristic reasoning*. Heuristic reasoning expands the scope of questions that one can answer with the same amount of quantitative knowledge. For example, Enrico Fermi famously asked his Physics class, “How many piano tuners are there in Chicago?” a question that highlights well the role of heuristic

reasoning and quantity sense. Most people, with the exception of people closely related to the piano tuning industry, are unlikely to have quantitative knowledge to directly answer the question. Yet, using heuristics like how many households own pianos, how often they might be tuned, and that the number of piano tuners can be estimated by first finding out just how many piano tunings might be needed, most people can relate it to quantities that they are likely to know. Figure 2.2 shows a schematic of this model.

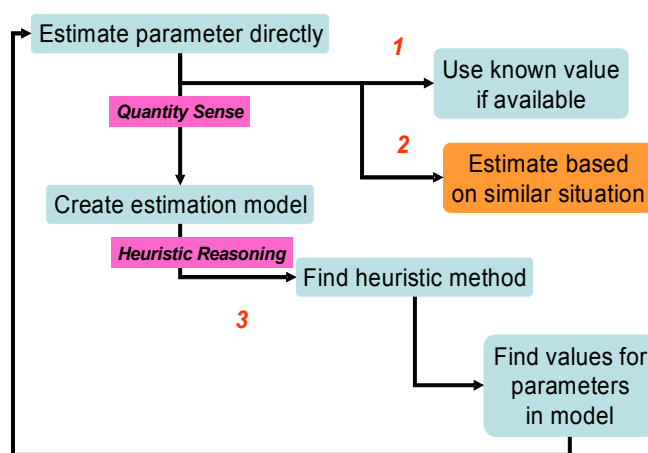


Figure 2.2: Quantity Sense and Heuristic Reasoning in BotE

The direct estimation step uses quantity sense, and the estimation modeling step uses heuristic reasoning. Below we describe these two steps in detail.

2.5.1 Direct Estimation

This involves directly estimating a parameter based on previous experience or domain knowledge. For instance, we might know the value of a physical constant, or use a value from a previous example that is highly similar to the current problem, or combine multiple similar

examples to estimate a value based on those prior values. Experience in a domain helps one develop their quantity sense for quantities in that domain: knowledge of similar values, typical values and a sense of the scale of values. The simplest type of direct estimation involves lookup. Humans good at estimation go beyond that and are able to use their quantity sense, specifically knowledge of typical values, scale (“1 Ampere is too high a current for a Walkman”) and similar experiences to generate an estimate when the answer is not directly available via lookup. Chapter 3 presents a cognitively plausible theory and model of this process of developing quantity sense from experiences. In this chapter we focus on estimation modeling.

2.5.2 Estimation Modeling

Estimation modeling is the process of building a simplified model of the situation that is good enough for the purposes of making a rough estimate. This is required when a parameter cannot be directly estimated. In such cases we build a model that relates the parameter in question to other parameters, which in turn must be either directly estimated or modeled. Estimation modeling is done by applying heuristic methods that are likely to yield subproblems that are easier to solve and can be combined to get an answer for the original question. For instance, in the piano tuners example above, multiplying the population of Chicago by the fraction of households that own pianos is a heuristic that can be used to estimate the number of pianos in the city. To estimate how many piano tunings happen every year, one might estimate how often a piano is tuned by using their experience of owning a piano, which is generalizing from a similar example, another heuristic method. In the next sections, we present a formalism for representing

heuristic methods for estimation modeling and a set of methods that we claim provide broad coverage for BotE reasoning.

2.6 Representation of BotE Problems

In this section we present our formalization of the BotE problems and heuristic methods. All BotE estimation problems ask for value of a quantitative property of some object. An abstract way to represent a BotE problem is $(Q \ O \ ?V)$ where Q is the quantity, O the object and $?V$ is the unknown value that is being sought. For example, in the question “How many K-8 teachers are there in the US?,” Q is cardinality, and O is the set of K-8 teachers in the US. Note that such decomposition of a question into a quantity-object pair is not necessarily unique, for example, “How much money is spent on newspapers in the US per year?” can be decomposed into:

Q =Cost, O =All newspapers sold every year in the US

Q =Annual sales, O =All US newspapers

Q =Annual newspaper sales, O =US

The answer to the question is the ground statement where $?V$ is bound to a numeric quantity and the justification. In case the answer was directly available in the knowledge base, then the justification is a reference to the knowledge base. However, when the answer was obtained by estimation modeling it consists of the dependency structure of the heuristic methods and subproblems solved along the way along with their justifications. The justification provides the structure for explanation and further exploration when the estimate generated is unexpected. An important aspect of BotE reasoning is that the same question can have multiple answers and

justifications. The numerical values of answers obtained using different heuristic methods will most likely differ, as each answer is an approximate estimate. Therefore $?V$ can have multiple bindings, each with its own justification. Let's consider the ground statement $(Q \ O \ V)$. Q is a binary predicate which attributes a numeric quantity to an object, e.g., height. V is a numeric scalar quantity, i.e., it consists of a number and a unit, e.g. 1.83 meters. In a well formed statement, the unit has to be in accord with the quantity Q . O can be a specific individual object, e.g., George Bush, or a collection of objects, e.g., adult male humans. In the latter case, V represents the typical value of Q for the collection⁶. In a more complex BotE question, O can represent a reified spatio-temporal chunk, and we can still use the simple representation proposed here for the question. Reification allows us to name a complex situation (like an event) and elaborate it using various roles that can be defined that are applicable in that situation [Davidson, 1967].

2.7 Representation of Heuristic Methods

Given a BotE problem $(Q \ O \ ?V)$, a heuristic method transforms it into a set of other problems $\{ (Q_i \ O_i \ ?V_i) \}$ such that $?V_i$ are already known or easier to estimate. Each heuristic method consists of:

- *Trigger conditions*: These specify the class of queries and additional conditions that must be true in order for this method to be applicable.

⁶ Alternatively, it could represent the range, distribution, or some compact summarized representation of the set of values that Q takes on for the various objects in the collection. Simple operations like mean or median, or more sophisticated mechanisms for selecting a typical value can be used in that case to transform it into our representation.

- *Subgoals*: These are the new subproblems, $\{ (Q_i \ O_i \ ?V_i) \}$, that this method proposes, which if solved, could generate an answer to the original query.
- *Result step*: This specifies how to combine answers found to the subgoals ($?V_i$) to generate the answer for the original query ($?V$).

With our problem representation, there are three kinds of heuristic methods based on what aspect of the problem it transforms:

- *Object-based*: $(Q \ O \ ?V) \rightarrow \{ (Q_i \ O_i \ ?V_i) \}$: An object-based method relates an object, O , to a set of objects, $\{O_i\}$, such that the quantity values for those objects, $\{?V_i\}$, combine in a known way to estimate the original quantity, $?V$. Note that since we are estimating the same quantity, this combination function can only be addition or subtraction since $?V$ and $\{?V_i\}$ have to have the same dimensions⁷.
- *Quantity-based*: $(Q \ O \ ?V) \rightarrow \{ (Q_i \ O \ ?V_i) \}$: A quantity-based method relates a quantity, Q , to a set of quantities, $\{Q_i\}$, such that the values of these quantities (for the object O) can be combined in a known way to derive the original quantity. Note that the combination function has to satisfy dimensional constraints [Szirtes and Rosza 1998], i.e., $?V$ and $f(\{?V_i\})$ have to have the same units, where f is the combination function.
- *System-based*: $(Q \ O \ ?V) \rightarrow \{ (Q_i \ O_i \ ?V_i) \}$: System-based methods transform both the quantity and the object into other quantities and objects. They represent relationships between quantities of a system as a whole. Most common

⁷ Some of the $?V_i$'s can be multiplied or divided by dimensionless quantities like ratios.

application of system based methods are when there is an invariant quantity, e.g., momentum that remains unchanged between different states.

The above breakdown is based on the syntactic transformations of the aspects of the question. In the next section we present seven heuristic methods: three object-based, two quantity-based, and two system-based, which we believe is a comprehensive set of heuristic methods for BotE reasoning. For each of the heuristic methods, we describe it, and then briefly discuss the sources of flexibility in the method.

2.7.1 Object-based Heuristic Methods

2.7.1.1 Mereology

Mereology (Lesniewski, 1916; Whitehead, 1919) is the study of parts and wholes. The basic primitive of mereology is the `partOf` relationship. BotE reasoning exploits this relationship to transform the object in question into other objects that are its parts or constituents. The values of quantities of subparts are systematically related to the value of quantity of the whole. An extensive quantity is a physical property that is dependent on the system size, for example, mass, volume, heat, etc.; while an intensive quantity is one independent of system size, for example, density, temperature, melting point, etc. If Q is an extensive parameter, then, $Q = \sum Q_i$. For example, the weight of a basket of fruits is the sum of the weights of all the fruits and the basket. This heuristic method requires making a closed world assumption, namely, that we know all the parts of the original object. In order for this method to be applicable, the parts should be non-overlapping along the quantity.

Many objects are distributed into subparts by a Zipf or Pareto distribution, e.g., 80% of the world population is contained in the twelve most populous countries. This is an ecological argument that allows us to extend the applicability of the mereology heuristic method to situations when all the subparts are not known. If \mathcal{O} is homogeneous, i.e., composed of the same kind of objects, then the above sum reduces to a product of the number of parts and the value for each part, $\sum \mathbf{v} = n * \Sigma \mathbf{v}'$. In some situations, homogeneity can be an assumption to approximate a more complex calculation involving all the subparts.

If \mathcal{Q} is an intensive parameter like density, we look for the constituents of \mathcal{O} . In this case, we need to know all the constituents and for each of them the fraction that they constitute of the whole, then, $\mathbf{v} = \sum \mathbf{w}_i * \mathbf{v}_i$, where \mathbf{w}_i is the fraction of the i th part. For example, the density of a mixture is the weighted average of the densities of the constituents. Once again, we can relax this method in cases when we do not know all the constituents, as long we know most of them, e.g., the density of human body is very close to that of water, as water is the largest constituent.

2.7.1.2 Similarity

The similarity heuristic method transforms the object into other object(s) which are similar to it. For example, when trying to estimate the rent for an apartment, a similar apartment in the same neighborhood whose rent is known is a reasonable guess. We call this value from the similar example as an *analogical anchor*. As a first pass, this analogical anchor is evaluated for its plausibility for the value sought. If two objects are similar, it doesn't warrant the inference that values of all the quantities for two objects are similar. For example, another grad student in my department probably gets paid similar to me, but doesn't necessarily weigh the same. Two

similar basketball players might have similar height, but two professors might not. This notion of what features can be inferred from a similar example was called *projectability* by Goodman (1955/1983). There is increasing psychological evidence that projectability is based on *centrality* of the feature [Ahn *et al*, 2000; Hadjichristidis *et al*, 2004]. A feature is central to the extent that features depend on it. In our above example, height is central to basketball players, but not to professors. We have operationalized this notion of centrality as the structural support of the inference in computation of similarity using the Structure Mapping Engine [Falkenhainer *et al*, 1989]. The structural support quantifies how causally connected that quantity is to other aspects of both the problem and the similar example.

Furthermore, the analysis of the comparison between the problem and the similar example provides the grist for computing adjustments from the analogical anchor to improve the estimate: for example, one might notice that the apartment that they were reminded of is smaller, and is in a slightly less desirable location. Let's assume that the domain knowledge about apartments contained the following causal relationships:

- A larger apartment has higher rent, all things being equal.
- The more desirable the location, the higher is the rent, all things being equal.

These facts suggest that the estimate of rent should be more than the rent of the reminded apartment. Just how much more? The effect of location on rent can vary, and in some neighborhoods, it might be stronger than others. At this point, one can use other examples to determine just how strong that effect is. We call these adjustments based on causal knowledge *causal adjustments*. The final estimate is generated by adjusting the analogical anchor to reflect

the causal adjustments. Chapter 3 gives more details about the representations and the algorithm that implement the similarity heuristic method.

2.7.1.3 Ontology

The ontology heuristic method tries to find other objects from the ontology hierarchy which might be used to guess the quantity in question. In the simplest form, if \mathbf{O} is an instance of \mathbf{O}_1 , then we can use the knowledge about the class to guess the value for the instance. For example, if we know that Jason Kidd is a point guard⁸, then we can use the knowledge that point guards are relatively shorter than other players on the team to guess his height. If we didn't have information about point guards, we could even use the fact that Jason Kidd is a basketball player to guess his height. As in the similarity heuristic method, the accuracy of the estimate is proportional to the centrality of the quantity.

Clearly, the further we are in the ontology from the original object, the less accurate will be the estimate. Here we describe the stopping strategy for the ontology heuristic method. Conceptually, the category hierarchy can be considered as specific \rightarrow subordinate \rightarrow basic-level \rightarrow superordinate [Rosch, 1975]. An example of the four levels of hierarchy will be a specific chair in my living room, armchair, chair, furniture. In this framework, we can see that not only the accuracy decreases as we go higher, but also that the basic level category is a good stopping point. Markman and Wisniewski [1997] have argued that the basic level category maximizes the within category alignable differences [Markman and Gentner, 1996]. The subordinates are too similar, and the superordinates have too few commonalities which decreases the number of

⁸ The point guard is one of the standard positions in a regulation basketball game. Typically one of the smallest players on the team, the point guard's job is to pass the ball to other players who are responsible for making most of the points.

psychologically relevant differences. Thus the basic level categories are defined as those that maximize the number of alignable differences. A much simpler stopping strategy is the variance heuristic: compute the variance of values of known instances of a category, if it is more than a threshold value, do not use this as an estimate.

2.7.2 Quantity-based Heuristic Methods

2.7.2.1 Density

The density heuristic method converts a quantity into a density quantity and an extent quantity. Here, density is used in a general sense to mean average along any dimension: we talk of electric flux density, population density, per capita income, etc. Rates, averages, and even quantities like teachers per student are examples of densities. For example, the number of K-8 teachers in the US can be estimated by multiplying the number of teachers per student by number of students. This heuristic method exploits the fact that many numbers and statistics are more readily available as densities. If an explicit density value is not available, for example, per capita income, then one can estimate it by looking at a typical example, or averaging a set of examples, or using the mereology strategy; depending upon the amount of knowledge available.

2.7.2.2 Domain Laws

This is a family of heuristic methods that uses domain laws to convert a quantity into other quantities. Domain laws include laws of physics as well as rules of thumb. For example, Newton's second law of motion, $\mathbf{F} = m \cdot \mathbf{a}$, relates the force on an object to its mass and acceleration. The application of domain laws by the problem solver requires formalizing the assumptions and approximations implicit in the laws. This has been well explored in

compositional modeling [Falkenhainer and Forbus, 1991; Nayak, 1994]. In BotE reasoning, aggressively applying approximations to simplify the problem solving becomes crucial. Some of the approximations are:

1. *Geometry*: Assume simplest shape, e.g. Consider a spherical cow [Harte, 1988].
2. *Distribution*: Assume either a uniform distribution, or a Dirac-delta (point mass).
3. *Calculus*: Integrals can be simplified by sums or average multiplied by extent, and differentials by differences.
4. *Algebra*: Use simplification heuristics to reduce the number of unknowns [Pisan, 1998].

2.7.3 System-based Heuristic Methods

System-based methods represent relationships between quantities of a system as a whole. They transform the quantity and the object in question into other quantities and objects simultaneously. It would seem that this effect can be obtained by sequentially applying a quantity-based and an object-based heuristic method (or vice versa) since all the above methods are compositional. There are two reasons for considering this as a separate type of heuristic method: 1) it represents a reasoning pattern that is different, 2) sometimes it is much more efficient to apply a system-based heuristic method, e.g., applying conservation of momentum leads to safely ignoring all the internal forces which need to be made explicit otherwise. The two system-based heuristic methods are described next.

2.7.3.1 System Laws

This class consists of physical laws that are applicable to a system as a whole. Many physical quantities remain conserved (or are “invariants”) for a system, e.g., energy, mass, momentum, angular momentum, etc. As a result of this, often one can write a balance equation that relates the expressions that denote the value of the quantity in two different states of the system. To avoid profusion of such balance equations between any two states, it is important to introduce them only if they relate mostly known parameters and introduce fewer new unknown quantities. In applying such a balance, appropriate assumptions about the system in consideration have to be made. For instance, while applying conservation of energy, we make a closed world assumption that there are no other sources or sinks in the scenario.

2.7.3.2 Scale-up

This is often an empirical heuristic method. A smaller model that works under the same physical laws can be used to estimate the quantity values for a full-scale system. To ensure that the scale-up is valid, all the dimensionless groups must be kept the same in the model and the prototype. For example, the Reynolds number is a dimensionless group that corresponds to the nature of flow (laminar, transient or turbulent), and for a flow model to be valid for scaling up, the Reynolds number must be the same in both situations.

2.7.4 Reasonableness and Comprehensiveness

Estimation models are constructed by applying heuristic methods to transform the problem. Heuristic methods are patterns of reasoning that yield *reasonable* answers and provide

comprehensiveness. Next we define reasonableness and comprehensiveness, which are analogous to notions of soundness and completeness in formal logic.

Reasonableness: What is reasonable varies depending upon the domain and task at hand. We said earlier that a factor-of-two accuracy is reasonable for BotE estimates. However, for many examples, like estimating the increase in Earth's temperature due to increasing greenhouse gases in the atmosphere over the next twenty years, we do not know the “correct” answer. Yet a reasonable estimate is quite valuable. In cases like these, reasonableness is established by providing valid justifications and showing that multiple approaches converge to a similar answer. The justification consists of facts, axioms, heuristic methods and their dependency structure used to arrive at the answer. Heuristic methods can be contrasted with domain specific laws like Newton's second law or Stokes law for computing drag force, where we are guaranteed a correct answer in the framework of that law's applicability. Heuristic methods are more broadly applicable at the cost being reasonable instead of accurate, allowing us to be able to answer questions that will be impossible to answer using domain specific laws. To summarize, the properties of a reasonable answer to a question are:

- It has a valid justification.
- It is similar to other answers to the same question with different justifications.
- It is similar to the correct answer to the question. (Optional, when answer available)

Comprehensiveness: The goal of BotE reasoning is to be able to produce a reasonable estimate quickly and despite missing knowledge. In the next section we present heuristic methods for answering BotE questions. The comprehensiveness of a set of heuristic methods is the fraction of all questions (of the class, e.g., all BotE questions) that they can be used to generate reasonable answers for. A comprehensiveness of 1 will indicate that we are guaranteed to answer any question of that class using the given set of heuristics. For open ended domains like BotE reasoning it might be difficult to establish the value of comprehensiveness. One approach might be to do an empirical analysis with a large sample of questions of that type to assess the comprehensiveness of the heuristics. Weak methods like hill-climbing and means end problem solving [Newell and Simon 1963] have high comprehensiveness with two caveats – 1) they assume a fully represented domain theory, and 2) they can take unbounded computational resources.

2.8 Corpus Analysis of Swartz's Back-of-the-Envelope Physics

BotE reasoning is very powerful because of its applicability to many domains. Clearly a human expert, or a computer problem solver, can do better with more quantitative and heuristic knowledge. The above analysis is an attempt to identify the core heuristic methods in BotE reasoning. In this section we present supporting evidence for the comprehensiveness of the heuristic methods.

We arrived at these heuristic methods by an analysis of the knowledge that our problem solver was using. To examine the comprehensiveness of this set of heuristic methods, we then manually went through Clifford Swartz's "Back-of-the-envelope Physics." Swartz's book is a

collection of estimation problems (along with solutions) from various domains in physics. Some examples of questions discussed in Swartz's book: How much heat is generated by an adult human? How much centrifugal force is experienced by a rider in a typical carousel? Even though the goal of our work is not tied to Physics, this provides an independent confirmation about our heuristic methods.

We looked at all the problems (n=44) from Force and Pressure, Rotation and Mechanics, Heat, and Astronomy. These domains represent a wide variety of problems. For every question, we first identify the quantity (Q) and the object (O). Then, we went through the worked out solution and extract all the quantities and objects mentioned. Every time a new quantity-object pair is introduced, we count that as a transformation which is classified as one of the seven heuristic methods or “other” category. Swartz's worked out solutions fortunately follow a similar scheme as the model presented above: he builds an estimation model, and plugs in the numbers and does the arithmetic as the last step. This model includes all of the transformations counted above, and is used for doublechecking them. We counted an application of the ontology strategy only if it was specifically mentioned, since trivially almost all problems apply that method, as most problems are not about specific instances but classes of things.

Table I shows the results of our empirical analysis. The table shows the number of instances of each heuristic method, and the corresponding percentage, for Swartz's book as well as for the set of problems that BotE-Solver can currently solve⁹.

⁹ The set of problems solved by BotE-Solver is different from those in Swartz's book. BotE-Solver's problems are from commonsense domains instead of Physics.

Table 2.1: Distribution of heuristic methods in Swartz's book.

Heuristic method ↓		Number of times used.	% of times used
H1	Mereology	11	14
H2	Similarity	5	6
H3	Ontology	6	8
H4	Density	10	13
H5	Domain laws	29	37
H6	System laws	11	14
H7	Scale-up	2	3
	Other	5	6
	Total	79	100

Some observations from table 2.1:

Coverage: The seven heuristic methods account for 94% of transformations in the problems from Swartz's book. The 5 in the "other" category contain four instances of designing experiments to estimate a quantity, and one instance of a complex problem from statistical mechanics. The experimental strategies exploit one or more of the seven methods, but there is considerable complexity in designing a good experiment. We focus on conceptual back of the envelope problems. The three syntactic possibilities for heuristic methods – object-based, quantity-based and system-based are complete. Furthermore, our analysis of problems from various domains and our problem solver that we will present in Chapter 4 leads us to conjecture that the seven heuristic methods are comprehensive for the task of back of the envelope problem solving.

Domain-specificity: Heuristic methods H1 through H4 are domain independent, and account for 40% of transformations, while 60% of transformations are domain specific, with the largest component being domain laws.

2.9 Summary

We presented a formalization for problems and heuristic methods for BotE reasoning. Using this representation, we described the three syntactic possibilities for types of heuristic methods: object-, quantity- and system-based. We then described the seven heuristic methods, which we claim provide broad coverage for BotE reasoning. The analysis of worked out problems from Swartz's Back-of-the-Envelope Physics book was presented to support this claim.

Chapter 3: A Theory of Quantity Sense

3.1 Introduction

Quantities are ubiquitous and an important part of our understanding about the world. Mileage of cars, the budget deficit of the country, heights of people, etc., are some examples of quantities. A facility for quantities, which we describe as *quantity sense*, is important for making decisions and arguments about why a course of action is better than another – for tasks ranging from buying groceries to choosing which policy to support. This chapter addresses the representational and computational questions underlying quantity sense: 1) What do people learn about quantities?, and, 2) How do people learn about quantities?

Our answers to these questions can be summarized as the *Symbolization By Comparison* (SBC) theory: People's knowledge about quantities consists of qualitative representations: symbolizations of the continuous quantity, which are built by the process of comparison. These symbolizations consist of named points and intervals on the scale of quantity that capture distinctions of quality, e.g., boiling point and poverty line, and distinctions of quantity, e.g., tall

and short. We present evidence from Psychology and Linguistics, and arguments from ecological and task/reasoning constraints that support the SBC hypothesis. We describe CARVE, a computational instantiation of the SBC theory: it learns qualitative representations of quantity from exposure to examples.

We present a functional evaluation of CARVE: we show that representations generated by CARVE lead to more accurate estimates in an analogical estimation task. Analogical estimation is using a similar example to make a numeric estimate. For example, the price of a used car might be similar to another car of the same make and mileage. In order to use analogies to make numeric estimates, our analogical matching algorithms should be sensitive to quantities in the first place. Most models of similarity do not adequately handle numeric properties – either ad hoc similarity metrics such as Euclidean distance are used, or the numerical values are completely ignored in the matching and retrieval processes. The SBC theory presents a different approach to the problem of incorporating quantities in similarity models by proposing that the solution lies in better representations, not in the similarity metrics.

The next section defines the basic terms used to describe quantity sense, and describes the space of quantitative knowledge. Section 3.3 presents a review of literature in education, linguistics, qualitative reasoning and psychology that bears on quantity sense. Section 3.4 presents the constraints on cognitively plausible representations of quantity. Section 3.5 goes into further detail about the distinctions that should be made in our representations of quantity. Section 3.6 describes CARVE. Section 3.7 describes analogical estimation, verbal protocols of experts on an estimation task, and KNACK, the computational model of analogical estimation. We then conclude in Section 3.8.

3.2 Quantity Sense

The notion of quantity is quite broad, and there is a substantial literature in psychology, linguistics and qualitative reasoning (QR) on many different aspects of it. We begin with a description of various terms used in describing quantities.

3.2.1 Definitions and Terminology

Descriptions of specific objects or abstract categories consist of *attributes* (also called *features* or *properties*), e.g., a description of dog might consist of size, color, pedigree, etc. The set of values that an attribute takes is called a *scale*. There are four types of scales:

- 1) *Nominal*: Attributes like gender, color, and ethnicity have values that belong to nominal scales, e.g., {male, female}. The only operation of comparison on a nominal scale is whether two values are same or different.
- 2) *Ordinal*: The values on an ordinal scale can be ordered, however, the differences between them are indeterminate, e.g., the scale of U.S.D.A beef ratings is {good, choice, prime}. The operations allowed on an ordinal scale are $>$, $<$, and $=$. It is not possible to add or subtract values. More specifically, the set of ordinal values are not closed under such operations if a fixed set of values is allowed.
- 3) *Interval*: The intervals between the adjacent values on an interval scale are equal with respect to the attribute being measured, e.g., Temperature in Fahrenheit is an interval scale: the difference between 32 F and 33 F is equal to the difference between 84 F and

85 F. Besides comparison, operation of addition and subtraction are allowed on these scales.

- 4) *Ratio*: When a scale has equidistant adjacent points (an interval scale), as well as a meaningful zero point (e.g., temperature in Fahrenheit does not – what value denotes the absence of temperature?), then it is a ratio scale. As a result, operations of multiplication and division are allowed on these scales. For example, income, or age are examples of ratio scales.

What scale might be used to describe an attribute is a function of one's knowledge and representation. For example, color values could be represented as any of the four scales above. We will call the attributes that take values on ordinal, interval and ratio scales as quantitative attributes, or, *quantities*. Quantities can be conveniently represented numerically, as numbers have all the properties to support the operations on all of the scales.

There are two types of quantities: *counts* and *measures*. The simplest type of quantity is counts, for example, the count of legs of a dog is four. Linguistically, count nouns are objects that allow modification by a number directly. Counts directly map on to whole numbers. Measures are more complex: there is no number directly ascribable to an amount of water, or a specific temperature. To understand measures, we need to understand the notions of *units* and *dimensions*. Dimension is a formalization of comparability of quantities. For example, wingspan of birds and height of people are different quantities and have different scales, but have the same dimension of length. Associated with a dimension is a unit of measurement, a standard that is agreed upon and reproducible by others. Measuring the amount of water involves counting the (real) number of units of the amount dimension, volume, in the given water. Being able to

measure also involves knowing the logic of measurement and operation of measuring tools like balances and scales of various kinds. Standardized units and dimensions are scientific conventions and are provided to most people via education.

However, we start learning about many of the quantities like size and weight on our own very early on [see for example, Smith, 1984]. By a combination of labels provided by language – dimensional adjectives [Bierwisch, 1967] like tall, heavy, expensive – we learn to identify quantities and abstract the scale of values that they can take. To reiterate, a quantity is an attribute whose space of values (or scale) has ordinal, interval or ratio scale properties.

3.2.2 The Space of Quantitative Knowledge

There are many types of quantities: directly perceivable, e.g., length; less directly perceivable, e.g., acceleration; conceptual, e.g., GDP and IQ; and subjective, e.g., spice-level in food. These quantities are involved in a variety of reasoning tasks: comparison, classification, estimation and arithmetic. Our knowledge about quantities is of various kinds: we understand that there are *Expensive* and *Cheap* things, that Canada is *larger* (in area) than the USA, that basketball players are usually *tall*, that the boiling point of water is *100 degrees Celsius*. Figure 3.1 shows some of the different research themes exploring questions in the space of quantitative knowledge. This picture might indicate more order and structure than there is in cognitive quantitative skills: we do not know what developmental and conceptual connections between various boxes in the figure. Each box in the figure concern a type of knowledge and/or reasoning process, and we mention seminal references that exemplify it.

Our knowledge of quantities is influenced by at least two broad directions: experiences and instruction. At the heart of our quantitative abilities is the core cognitive infrastructure for representing counts and continuous magnitudes, shown in Box 1 in the figure. Although there is disagreement between theoretical interpretation and details, it is well established that humans (along with many other animals) possess a language independent, non-verbal ability to count small numbers (1/2/3 and sometimes 4), and approximately estimate large numerosities [Dehaene, 1999; Gallistel and Gelman, 2000]. This has been called *number sense* in this literature¹⁰. On the education side, mathematics provides powerful abstractions: numbers (reals and integers) to describe quantities. Knowledge of units and dimensions, and causal laws enable powerful qualitative and quantitative reasoning.

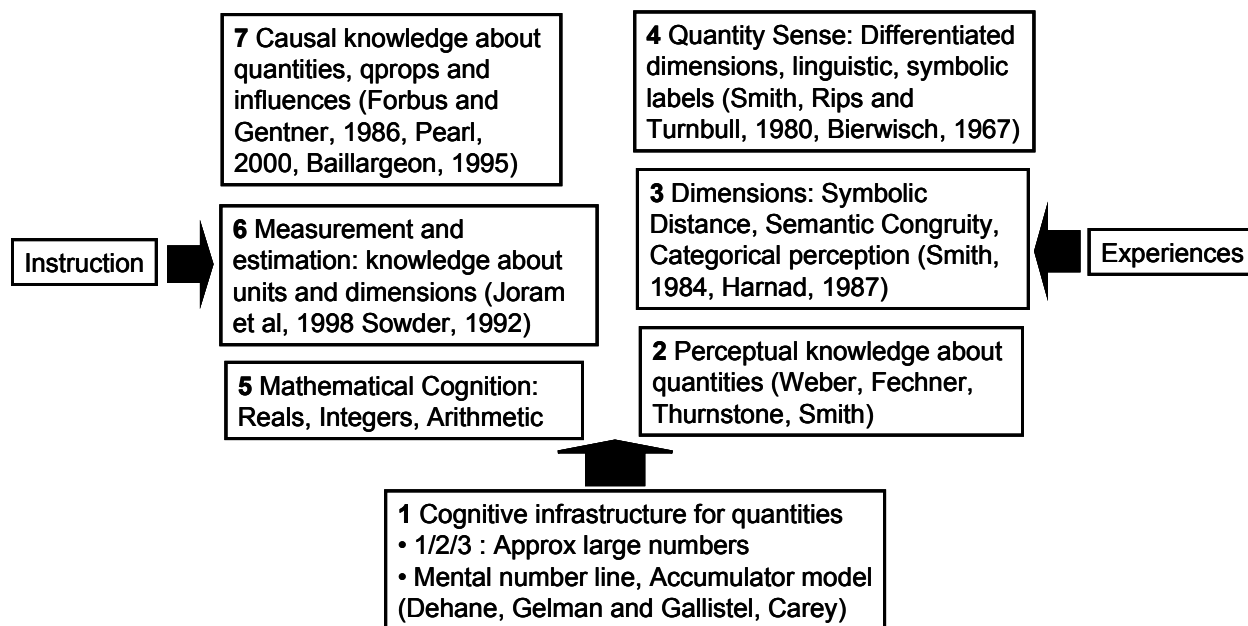


Figure 3.1: A schematic of a subset of space of research on quantity in Cognitive Science.

¹⁰ Math educators have included other quantitative skills like measurement, estimation and arithmetic when using the phrase “number sense,” sometimes using it in place of the more popular “numeracy.” [Madison and Steen, 2003 (Quantitative Literacy)]

On the experiences side, we begin with perceptual knowledge of sizes, weights, colors, etc (2), and slowly extract more abstract dimensions. We further build qualitative abstractions of continuous quantities. Some examples include *tall* and *short* for the quantity of height of people; *poverty line*, *lower class*, *middle class* and *upper class* for income of people; *freezing point* and *boiling point* for the temperature of water. We will refer to such qualitative abstractions of the continuous as symbolizations. Expertise in quantitative domains consists of building and mapping symbolizations on to the scale of quantities: having a sense of reasonable, high and low values, and causally significant points and intervals. We will refer to this set of skills and learning mechanisms as *quantity sense*.

These symbolizations and their mapping onto quantitative values seem to be determined by a mixture of personal experience (e.g., what I consider to be *spicy* in regards to food), society (e.g., *middle class*), science (e.g., phase transitions). Some are task-specific – one makes more distinctions than freezing and boiling for bath water. Furthermore, some of these symbolizations have been said to be vague [Varzi, 2003], in the sense that it is not possible to tell exactly at what value of height one becomes tall. Given these concerns, there is little work addressing the issue of finding systematic principles behind such symbolizations. In this chapter, we take a closer look at what our representations of quantities contain, guided by cognitive and linguistic evidence, and ecological constraints on our knowledge of quantities. We address the following two fundamental questions about people’s knowledge of quantities –

- *Representational*: What do our representations of quantity look like? Or, what representational machinery is needed to make the distinctions that we do?
- *Computational*: How are these representations built with experience?

These are questions about how cognition works, as well as about how the world is organized. To quote William James (1890), “The components of an absolutely changeless group of not-elsewhere-occurring attributes could never be discriminated. If all cold things were wet, and all wet things cold; is it likely that we should discriminate between coldness and wetness?” This is opposed to Bierwisch (1967), who argued that dimensional adjectives do not represent “properties of surrounding world in the broadest sense, but rather certain deep seated properties of the human organism and the perceptual apparatus.”

This distinction is important, as many of representations of quantity (for example, in scientific domains, and thus those in Qualitative Reasoning) tend to have the underlying perspective of representing properties of the world in some optimal fashion. The `Boiling Point`, for example, seems to be more a property of the world than the notion of `Expensive`, which seems more variable and harder to formalize. For example, consider the quantity temperature which has associated adjectives like cold, tepid, lukewarm, warm, and hot. We argue that those linguistic distinctions play an important role in our representations of quantity and their development. The quantity space representation [Forbus, 1984] has the expressive power that our representations of quantity seem to have, and we extend and provide cognitive and linguistic evidence in its support. The quantity space representation provides an important insight: only the *necessary and relevant* distinctions should be made. However, it does not provide us with an algorithm to automatically discover those necessary and relevant distinctions, which is the subject of section 3.4.

3.3 Background and Motivation

This section presents a background of relevant research from education, linguistics, psychology and qualitative reasoning. We begin with high-level motivation coming from mathematics education. Since we evaluate our theory in the context of analogical estimation, we present deficiencies in the current models of similarity, retrieval and generalization.

3.3.1 Education

One of the goals of an education is to instill quantitative knowledge and reasoning required in everyday life of an informed citizen. On one hand, we have increasingly increased amount of quantitative information available – “the world of the twenty-first century will be a world awash in numbers” [Steen, 2001]. On the other hand, studies testing quantitative knowledge and skills portray a dismal picture [Corle, 1960, 1963; Swan and Jones, 1971, 1980; Sowder, 1992; Linder, 1999]. Cockroft [1982] was one of the first to popularize “numeracy” as an important educational goal, while “innumeracy” seems to be the state of affairs [Paulos, 1988]. An important contribution of the research from mathematical education has been the identification and organization of various quantitative skills: counting, measuring, estimating, mental computation, and arithmetic, to name a few. Many labels, for example, numeracy [Cockroft, 1982], quantitative literacy [Porter, 1997], empirical mathematics [Packer, 2001], and number sense [Greeno, 1991] have been used to describe different combinations of knowledge and reasoning abilities related to quantities.

3.3.2 Linguistics

In language, one of the ways in which symbolizations get represented is by relative adjectives like `large` and `tall`. Relative adjectives are different from absolute adjectives like `rectangular`, `red` and `married` in the sense that (1) they can imply varying degrees of the property in question, as opposed to all-or-none for the absolute adjectives, and (2) their meaning varies with context, e.g., `tall` means different things in context of people and buildings.

These adjectives have been variously called degree, relative, gradable or dimensional adjectives [Bierwisch 1987]. Here we will stick to the term dimensional adjectives, emphasizing our focus on those that denote quantity. It has been proposed that dimensional adjectives denote measure functions that maps from objects to quantity values/ intervals [Kennedy, 2003]. It has long been recognized by linguists that dimensional adjectives convey an implicit reference to a norm or a standard associated with the modified noun [Sapir, 1944]. This implies two steps in interpreting a phrase like “a large x ”: (1) x establishes a *comparison class*. A comparison class is a set of objects that are in some way similar to x . For instance, in some cases, this comparison class might be the immediate superordinate of the subject [Bierwisch, 1971, see Vogel, 2004, for an analysis of Swedish dimensional adjectives]. How to obtain the comparison class is an open question. Staab and Hahn (1998) propose a computational model that uses knowledge about correlations to determine comparison classes on the fly. (2) Once the comparison class has been found, a *standard of comparison* is computed for the class. It is usually believed that this is the norm value of the property for the comparison class, but Kennedy (2003) observes that it can also be the minimum or maximum (e.g., full and open).

An important observation is the pervasiveness of symbolized distinctions of quantity across languages. Even languages and cultures with very limited numeric vocabulary, for example, the Pirahã tribe in Amazonia [Gordon, 2004] and Mundurucu [Pica et al., 2004], an Amazonian language have the equivalent notions of “small” and “big”¹¹. In contrast, such representations are not very common in formalizations of natural sciences. Knowing that the water is “hot” does not necessary sanction application of any physical law (while knowing that the water is hotter than the atmosphere, or at a temperature more than boiling point, does). We believe that the pervasiveness of dimensional adjectives in language is because of their role in processes of comparison: being large makes two things similar in the same way being red does. Discovering phase transitions might be a big scientific achievement, but words like “tall” and “hot” are a compressed record of many comparisons, and help setup implicit ordinal relationships in descriptions.

Two major issues that are little addressed in this literature are: 1) There is no general account of the comparison set¹², and 2) There is no concrete account of how the norm or standard of comparison for a set is computed. In cases where we are referring to stable taxonomic categories like insects and countries, it is believed to be some kind of central tendency. But clearly, it is more than a central tendency, since that would imply that most things in this world will be either large or small, as not many will be exactly equal to the norm.

¹¹ Both the works cited were focused on exploring the concept of number and arithmetic. However, the notion of quantity is different from that: one might have a very good notion of how heavy things are (a quantity), without being able to describe that in any numeric terms.

¹² Consider: 1) seeing your nephew after a year during which he had a growth spurt and exclaiming, “You are tall!” 2) or the disclaimer on coffee cups in the US, “Caution: the contents are hot.” In both these cases comparison sets are not taxonomic.

3.3.3 Qualitative Representations of Quantity

One of the goals of qualitative reasoning research has been to understand human-like commonsense reasoning without resorting to the preciseness of models that consist of differential algebraic equations and parameters that are real-valued numbers. There is a substantial body of research in QR that has shown that one can, indeed, do a lot of powerful reasoning with less detailed and partial knowledge [Forbus, 1996]. Qualitative reasoning has explored many different representations: status algebras (normal/abnormal); sign algebra ($-$, 0 , $+$), which is the weakest representation that supports reasoning about continuity; quantity spaces, where we represent a quantity value by ordinal relationships with specially chosen points in the space; intervals and their fuzzy versions; order of magnitude representations; finite algebras, among others. The representations differ in the *kind* of distinctions that they allow us to make. While these representations are very promising for cognitive modeling, there has been little psychological work to date on this. To echo the questions raised in the introduction, we are interested in finding a cognitively sound representational framework for these distinctions, and principles for finding the distinctions that we do and should make.

Our answer to the first question raised in the introduction is that the quantity space representation, augmented with distributional information, accounts for observations and existing evidence from psychology and linguistics. Our answer to the second question is the first attempt to come up with a general theory of what distinctions to make. Sachenbacher and Struss (2000) attacked a similar problem. They were interested in finding the right distinctions given the reasoning task. Here we are more concerned with cognitively plausible distinctions – for

example, the distinctions that are made in natural language on the space of values that the quantity takes.

3.3.4 Relevant Psychological Phenomena

3.3.4.1 Context sensitivity

Rips (1980) considers two hypotheses about how absolute and relative adjectives might be stored in memory – Pre-Storage and Computational model. For absolute adjectives like `married` and `pink`, he accepts the pre-storage model, where these predicates are stored with the concept they apply to. But because of context dependence of relative adjectives like `big`, e.g., in, “Flamingos are big”, he argues against storing these predicates in memory. We might have a predicate `pink` attached to `flamingo`, but in order to decide a flamingo is larger than an eagle, we might need a predicate `is-larger-than-an-eagle` associated with `flamingo`, which then deescalates into having infinitely many of those like `is-larger-than-turnips` and so on. He also observes that relative adjectives don’t propagate in a `isa` hierarchy – e.g., Grasshoppers are large insects does not imply Grasshoppers are large animals, but if you replace ‘large’ by ‘green’, the implication is right. He then shows reaction time and error rates for verifying the truth of statements containing relative adjectives which supports a different model. In his ‘computational model’ no relative information is stored. Attached to every predicate is a normal value, e.g. with `insects`, a normal size of quarter inches. An object is called large if it is bigger than this normal size. Once again the problem is that just storing the norm doesn’t tell you when the object can be

classified as large. The representation that we propose in section 3.4 solves his concerns with pre-storage models.

3.3.4.2 Reference Points and Categorization Effects in Comparison

The psychological reality of reference points on the scale of quantity has been shown in various domains. Rosch (1975) argued for the special status of such “cognitive reference points” by showing an asymmetry – namely that a non-reference stimulus is judged closer to a reference stimulus (e.g., the color off-red to basic-red) than otherwise, while such relationship between two non-reference stimuli is symmetric. Landmarks are used to organize spatial knowledge of the environment which exhibit similar asymmetries [Holyoak and Mah, 1984, among others]. Other relevant psychological studies that support the existence of reference points come from categorical perception [Harnad, 1987] and sensitivity to landmarks [Cech and Shoben, 1985]. Brown and Siegler (1993) proposed the *metrics and mappings* framework for real-world quantitative estimation. They make a distinction between the quantitative, or metric knowledge (which includes distributional properties of parameters), and ordinal information (mapping knowledge).

3.3.4.3 Models of Retrieval, Similarity and Generalization

Models of similarity and retrieval in case-based reasoning [Ashley, 1990; Leake, 1996; Ram and Santamaria, 1997] use numeric information, but they employ *ad hoc* similarity metrics such as Euclidean distance that are not psychologically grounded. In the domain of estimation of length of software projects, ANGEL [Shepperd and Schofield, 1996] makes an estimate by retrieving a

similar project. The similarity is defined by a numeric error metric that is minimized. However successful some of these applications might be, we don't think they tell us much about how numeric quantities are implicated in similarity judgments.

The structure-mapping engine (SME) [Falkenhainer *et al*, 1989] is a computational model of structure-mapping theory [Gentner, 1983]. Given two structured propositional representations as inputs, the *base* (about which we typically know more) and a *target*, SME computes a *mapping* (or a handful of them) between the representations. MAC/FAC [Forbus *et al*, 1995] is a model of similarity-based retrieval that uses a computationally cheap, structure-less filter before doing structural matching. It uses a secondary representation, the content vector, which summarizes the relative frequency of predicates occurring in the structured representation. The dot product of content vectors for two structured representations provides a rough estimate of their structural match. SEQL [Kuehne *et al*, 2000] provides a framework for making generalizations based on computing progressive structural overlaps of multiple exemplars.

One limitation of these models – and of other models of analogical processing (e.g., ACME [Holyoak and Thagard, 1989, LISA [Hummel and Holyoak, 1997], ABSURDIST [Goldstone and Rogosky, 2002]) – is that they do not handle numerical properties well. In all these models, numbers are treated like symbols, so 99 and 100 are as similar/different as 99 and 10000. When treated as symbols, they are both non-identical symbols, but numerically, the differences in magnitude are quite different. As a consequence, we have the following limitations in the retrieval, matching and generalization processes:

Retrieval: Just as Red occurring in the probe might remind me of other red objects, a bird with wing surface area of 0.272 sq.m. (that is the Great black-bucked gull, a large bird) should remind

me of other large birds. This will not happen in the current model, unless we abstract the numeric representation of wing-surface-area to a symbol, say, *Large*. That will show up in content vectors and thus contribute to retrieval.

Similarity: : Similarity between two quantities should be computed and combined together in a cognitively plausible fashion, which amounts to answering: 1) How to compute similarity along a quantitative dimension? And 2) How to combine similarities along different quantitative dimensions? For example, in current matchers, two cars which have identical values for all dimensions have the same similarity as two that differ in some dimensions, if other aspects of their representations are identical.

Generalization: A key part of learning a new domain is acquiring the *sense of quantity* for different quantities. E.g., from a trip to the zoo, a kid probably has learnt something about sizes of animals.

None of the above constraints are upheld in SME, MAC/FAC and SEQL. A large part of this deficiency, we feel, is due to poor representations of quantity. A symbolic and relational representation of the kind we propose here would make these models more quantity-aware.

3.4 Representing Quantity Sense

In this section, we present and argue for a cognitively plausible representation of quantity. There are three subsections: 4.1 organizes arguments for what must be contained in our representations of quantity around various constraints, 4.2 presents the proposed representation, and 4.3 discusses some implications of this representation.

3.4.1 Constraints

A representation of quantity allows us to make certain distinctions. Real numbers allow us to make too many, and dividing the range of values into three equal sized parts doesn't necessarily provide useful distinctions. Representations do not arise in vacuum – they are molded by the kinds of reasoning tasks we perform with them (reasoning constraints), the underlying reality of the things we are trying to represent (ecological constraints), and how we perceive this reality (psychological constraints). Based on these, and scattered pieces of evidence from psychology and linguistics, we argue that our representational machinery for quantities must contain partially (or possibly totally) ordered symbolic reference points (*a la* quantity space), and distributional information about the quantity (or an informational equivalent thereof).

3.4.1.1 Reasoning Constraints

The three distinct kinds of reasoning tasks involving quantities are:

1. Comparison: These involve comparing two values on an underlying scale of quantity (or dimension¹³), e.g., “Is John taller than Chris?” Our knowledge of how the quantity varies (its distribution), and linguistic labels like `Large` and `Small`, are but a compressed record of large number of such comparisons. The semantic congruity effect [Banks and Flora, 1977] is the fact that we are better and faster at judging the larger of two large things than the smaller of two large things – e.g., subjects are faster and more accurate at interpreting “A whale is larger than an elephant” than “An elephant is smaller than a whale.” Based on experiments involving adults

¹³ Consider “The space telescope is longer than it is wide.” These cross-dimensional comparisons can get quite complicated to interpret, e.g., “The Sears tower is as tall as the San Francisco Bay Bridge is long” does not literally mean that `Height(Sears Tower) <= Length(San Francisco Bay Bridge)`. See Kennedy (2001) for an analysis and implications of such comparisons.

learning novel dimension words, Ryalls and Smith (2000) suggest that in usage, we make statements like “X is larger than Y” more often than “Y is smaller than X”, if X and Y are both on the large end of the scale. This asymmetry in usage is a partial explanation of the semantic congruity effect.

2. Classification: These involve making judgments about whether a quantity value is equal to, less than or greater than a specific value¹⁴, e.g., Is the water boiling?, Will this couch fit in the freight elevator?, Are they below the poverty line?, etc. Usually, such classifications involve comparisons with interesting points on the space of values that a quantity can take, moving across which has consequences on other, different aspects of the object in concern. The metaphor of *phase transitions* describes many of such interesting points, although such transitions in everyday domains are not as sharply and well defined as in scientific domains (consider Poverty line versus Freezing point). We talk about this more in the next section.

3. Estimation: These involve inferring a quantitative/numeric value for a particular quantity, e.g., How tall is he? What is the mileage of your car? This is the activity that has the strongest connection to quantitative scales – one can go a long way in accounting for the above phenomena without resorting to numbers, but estimation involves mapping back to numbers [Joram et al. 1998]. Knowledge of interesting points on the scale might play an important role in estimation, for example in providing anchors to adjust from [Tversky and Kahneman, 1974].

These tasks are not completely distinct from each other – classification involves comparison, and estimation might be used in the service of classification. Two constraints on cognitively plausible representations that follow from the above tasks:

¹⁴ Or corresponding judgments involving intervals.

1. Our representations must keep track of interesting points on the scale of quantity, to classify, as well as to estimate.
2. Labels like `large` aid in making comparisons, as they setup implicit ordinal relationships (it is larger than the expected norm), which seem to be references to the underlying distribution of the quantity values.

3.4.1.2 Ecological Constraints

Quantities vary in a different fashion than nominal attributes. Our representational framework must be capable of capturing the interesting ways in which a quantity varies in real-world instances of it. Here are two different kinds of constraints on values a quantity can take –

1. Distributional Constraints: Most quantities have a range (a minimum and a maximum) and a distribution that determines how often a specific value shows up. For example, the height of adult humans might be between 4 and 10 ft, with most being around 5-6.5ft. `Tall` and `Short` refers to the underlying distribution of heights of people. A popular account of dimensional adjectives (e.g., “Flamingo is a `large` bird”) is that it establishes a comparison to an underlying categorical norm [Rips, 1980; but see Kennedy, 2003]. But it seems more than just reference to the norm; anything greater than the norm is not `large` or `high` – it also implicitly takes into account the spread of the distributions. More than just the norm, we can usually talk about the `low`, `medium`, `high` for many quantities, which seems to be a qualitative summary of the distributional information. There is psychological evidence that establishes that we can and do accumulate distributions of quantities. We describe the most compelling of such studies here –

refer to Peterson and Beach, 1967; Fried and Holyoak, 1984; Kraus *et al*, 1993; Ariely, 2001, among others, for more. Malmi and Samson (1983) presented subjects with one hundred three-digit numbers, which they were told were SAT scores of two different groups (named PIM and DAP). Each “SAT score” was displayed as either of PIM or DAP, and the three-digit number. Even when the numbers were displayed for merely 0.5 seconds, subjects accurately estimated (within 95% confidence interval of the stimulus mean) the mean for both PIM and DAP samples in the case of normal, skewed distributions and bimodal distributions. The last one suggested that subjects might be storing more than just a running mean, and so the experimenters tested the subjects for how accurately they could reproduce the entire frequency distribution of the sample. The subjects were able to reproduce the distributions qualitatively, as well as quantitatively. Surprisingly, the next question of how we partition these distributions has not been raised at all¹⁵.

2. Structural Constraints: A quantity is also constrained by what values *other* quantities in the system take, its relationship with those other quantities, the causal theories of the domain; in general, the underlying structure of representation¹⁶. For instance, for all internal combustion engines, as the engine mass increases, the Brake Horse Power (BHP), the Bore (diameter) and the Displacement (volume) increases, and the RPM decreases. These constraints represent the underlying mechanism, or the causal model of the object. As we move along the space of values a quantity can take, it is possible that we transition into a region where the underlying causal story is different (e.g., ice starting to melt, at the freezing point), which induces extremely

¹⁵ Fuzzy variables [Zadeh 1965] can take on ‘linguistic values’ like Large, Medium and Small; and allow us to represent overlapping range of values for these symbols. However, the specific mapping of Large-ness to area of countries, for instance, is a choice of the person building the representation, and is not in the scope of fuzzy logic. Our focus here is that mapping.

¹⁶ Comic books, mythology, and fantasy, for example, have the freedom to relax this constraint – a character can be arbitrarily strong, large, small or be able to fly, even though the physical design of the character might not be able to support it.

important and interesting distinctions of *quality* on the space of quantity. Much of the representations in QR involve such transition points.

These two ecological constraints point us to the two different kinds of information about quantities which must be parts of our representations:

1. Distributional information about how the quantity varies.
2. The quantity's role in, and relationship to, the underlying structure/mechanism, and the points at which there are changes in underlying structure.

3.4.2 Proposed Representation

Based on the observations in section 3.1, here we propose that our representations must contain symbolic reference points and distributional information.

3.4.2.1 Symbolic references to quantity

A partially, or possibly totally ordered set of symbolic reference points forms the *quantity space* [Forbus 1984]. Any value on the scale can then be represented via ordinal relationships to these symbolic reference points. Quantity space is the minimal representation that supports variable resolution. The symbolic and relational nature of this representation automatically makes it much more useful in our (structured/symbolic) representational framework. In the original formulation of quantity space, these symbols are limit points, those points where different processes/model fragments become active or de-active. We will relax that constraint in the discussion to follow and see what other kind of reference points are needed. The two main types of symbols in our quantity spaces are:

1. **Distributional Partitions**¹⁷: Symbols like `Large` and `Small`, which arise from distributional information about how that quantity varies.
2. **Structural Limit Points**: Symbols like `Boiling Point` and `Poverty Line`, that denote changes of *quality*, usually changes in the underlying causal story and many other aspects of the objects in concern.

Distributional partitions manifest as intervals centered around a norm, and structural limit points as boundaries demarcating transitions. Dimensional adjectives like `large` depend upon the context. Consider area of African countries: let's say Algeria is large, Swaziland is small, and Kenya is medium sized. We represent this as follows –

```
(isa Algeria
  (HighValueContextualizedFn
    Area AfricanCountries))
```

`High/Medium/LowValueContextualizedFn` are functions that take two arguments: a quantity and a context argument, a collection of objects. So in the above example `HighValueContextualizedFn` denotes the collection of large African countries, and the `isa` statement says that Algeria is an instance of that collection. The `LowValueContextualizedFn` similarly lets us represent the negative end, for instance small and cheap.

¹⁷ Interesting asymmetry here – Most of the distributional information is symbolized as intervals, and not points.

3.5 Necessary, relevant, and more distinctions

In this section we describe structural limit points and distributional partitions with examples.

3.5.1 Structural Limit Points

Structural limit points are a generalization of the idea of limit points introduced in QP theory. One should only make the *necessary* and *relevant* qualitative distinctions, QP theory advises us [Forbus, 1984]. In the domain of processes, QP theory provides the intuition for these distinctions: *where things change*, i.e., different processes and/or model fragments get de/activated, e.g., Freezing Point and Boiling Point of a liquid. Is there a general principle that provides these distinctions for more than just dynamical processes?

One can always partition the quantity space arbitrarily – so, one could have an ad hoc rule that said that we’ll always divide the space between the minimum and maximum into three parts – high, medium and low¹⁸. We are suggesting that there are some partitions that are more *natural* than others. Some features of the natural partitions –

Right level of granularity: Freezing Point and Boiling Point might be fine for reasoning about physical behavior, but if one is talking about shower water, then more distinctions like Cold, Body Temperature, Warm and Scalding Hot might be more appropriate.

Structurally predictive: of other properties of the system, e.g., Poverty Line, Lower Class, Middle Class, Upper Class.

¹⁸ For example, the Fuzzy logic community does something in the same spirit.

The *structural constraints* on quantities reflect a fundamental fact about the way things are in the world. Things in the world come in *packages* or *bundles*. For example, a “muscle car” has a powerful engine, is expensive, is designed for style and fun rather than safety or practical driving. In psychological literature, a similar notion is expressed by *attribute co-variation* or *feature correlation* [Malt and Smith, 1984; Kersten and Billman, 1992 and McRae, 1992]. But there’s much more than that – these are not merely bundles of correlated attributes, but are *structural bundles*. The entities, and quantities associated with them, tied by relations and higher order relations constraining them, give rise to the structure¹⁹ therein. Processes (as in QP theory), are a special case of these *structural bundles* (where the key relationships are of causality and influence) for the class of dynamical physical systems. Thus, the key idea is:

The necessary and relevant qualitative distinctions correspond to discontinuities in the underlying reality as captured by the structure in the representation.

Let us look at an example – consider people’s income. Poverty line, lower class, middle class and upper class define changes of quality on the space of income, as we expect that many other aspects of people – their lifestyle, the amount of time/money they spend on entertainment, education, the kind of vacations they have (or do not), the family and social climates in which they live, their expectations and relationships to the rest of the social structure, among other parameters, changes as we move across these interesting partitions of the scale of income.

Consider the size of dictionaries (as measured in number of pages, volume, or weight). There seem to be at least three meaningful distinctions of quality that might be projected on to size – pocket, table-top, and library-sized dictionaries. The tradeoffs for these three types of dictionaries are quite different. The key aspect of the pocket dictionary is portability, and thus it

¹⁹ The structure of relationships is an even more general notion than causality, spatial arrangement, connectivity.

has finer print, thinner pages, less detailed meaning, probably not much etymology and usage information, etc; the key aspect of the library sized one is comprehensiveness, and thus it follows that it is larger, heavier, has a much higher number of entries and even arcane and obsolete words, etymologies, usage information, is well bound as it is big and thick, has pages that are tougher so as to stand more usage, etc. The table-top dictionary falls somewhere in between. On the dimension of size, thus, the distinctions of pocket, table-top and library-size define interesting distinctions which have deep relationships to the underlying causal model, the underlying quality of dictionaries.

These changes of quality in the above two examples are reminiscent of phase transitions in physics/thermodynamics. In the same way as phase transitions, a set of underlying properties and the relationships that tie them together change as we move across the structural limit points. There are two types of phase transitions: first-order (sharp discontinuity, e.g., solid→liquid change), and second-order (where one can continuously move from one phase to another, e.g., magnetization)²⁰. The structure of relationships is the analogue for equations of state that hold in a particular phase, and the crisp/soft distinctions are analogous to first-/second-order transitions.

3.5.2 Distributional Partitions

The importance of the structural limit points presented in the previous section is apparent – they are predictive of structural properties of the system, and thus quite useful in doing qualitative reasoning. Surprisingly, though, the language contains many references to quantity which look very different from the structural limit points or the intervals they might imply. Consider

²⁰ See Sethna, 1992 for an introduction, and Gunton et al, 1983 for more detailed explanation

Large, Tall, Short, Expensive, etc. When we say a large flamingo, that is a reference to the distribution of sizes of flamingos and the fact that the particular flamingo we are looking at is larger than the norm. Such distinctions like small, medium, large, seem to be making cuts based on the distribution of values that the quantity takes, and the four most common distributions – uniform, normal, skewed, and Zipf, have different intuitions. An intuitive understanding of the normal distribution might be that there are fewer short and tall people than there are people of regular height (and also that the range of tall and short is larger than the regular size). The power law, or the Zipf distribution is an interesting case interesting as a meaningful norm for such distributions can not be defined. Are there some systematic ways in which people make cuts on a distribution they have abstracted? There is little known about this.

3.5.3 Implications

Most symbolic references to quantity have both a structural and distributional interpretation of them – so being Tall has structural consequences, for example, for a basketball player (or a gymnast). An interesting issue is the interactions between these two types of partitions. When do we choose to use structural partitions, and when distributional? The answer has to do with the nature of the quantity. Some quantities are more causally central – i.e., more deeply affect other aspects of the system than others (compare horsepower of a car to size of the door handles). In the class of examples that we are looking at, there will be a tendency to describe a quantity purely using distributional information if –

- The parameter doesn't have deep causal connection to the rest of the system, or it is not causally central (in terms of structured representation, has low *systematicity*), e.g., height of poets as compared to height of basketball players.
- There is not much of variation in the underlying structure (as far as is known in our representation) at all, e.g., size of adult male penguins.

Informal analysis of symbolic references to quantity in natural language provides support for the representation proposed above in language, but there are differences.

Prevalence of distributional partitions over structural limit points: Language is full of dimensional adjectives like hot, cold, etc., which usually are distributional partitions; as opposed to structural limit points²¹, which are the kind we find in scientific domains, and thus QR. When one begins to learn a domain, the distributions are accumulated until we know enough to give them symbolic labels, and the distributional partitions then helps us build the causal structures that then lead to the structural limit points. Dimensional adjectives also allow for flexibility in their usage and interpretations, making them linguistically useful. Distributional distinctions typically manifest as intervals, whereas structural distinctions typically are found as points. A plausible conjecture is that intervals are more informal and let us talk about the quantity without making commitments to where exactly the transitions happen.

Crisp versus soft structural limit points: Structural limit points are less crisp in everyday domains as compared to scientific domains. For example, the lines dividing the table-top and

²¹ This is true even when the transitions are sharp, e.g., dry → wet.

library-sized dictionaries, or the middle and upper classes, are less crisp than the freezing point. One plausible reason is the multiplicity and subjectivity of causal models in everyday domains, whereas, the equations of state that determine the scientific causal models are simpler and better described.

3.6 CARVE: Symbolization by Comparison

In the previous sections we argued that symbolization of continuous quantities is a key part of quantity sense. The question we now address is how these symbolizations are learned from experience. This question can be further broken down into:

1. How do we know which quantities to compare in order to build a scale?
2. How are the distributional partitions and structural limit points computed from these scales?

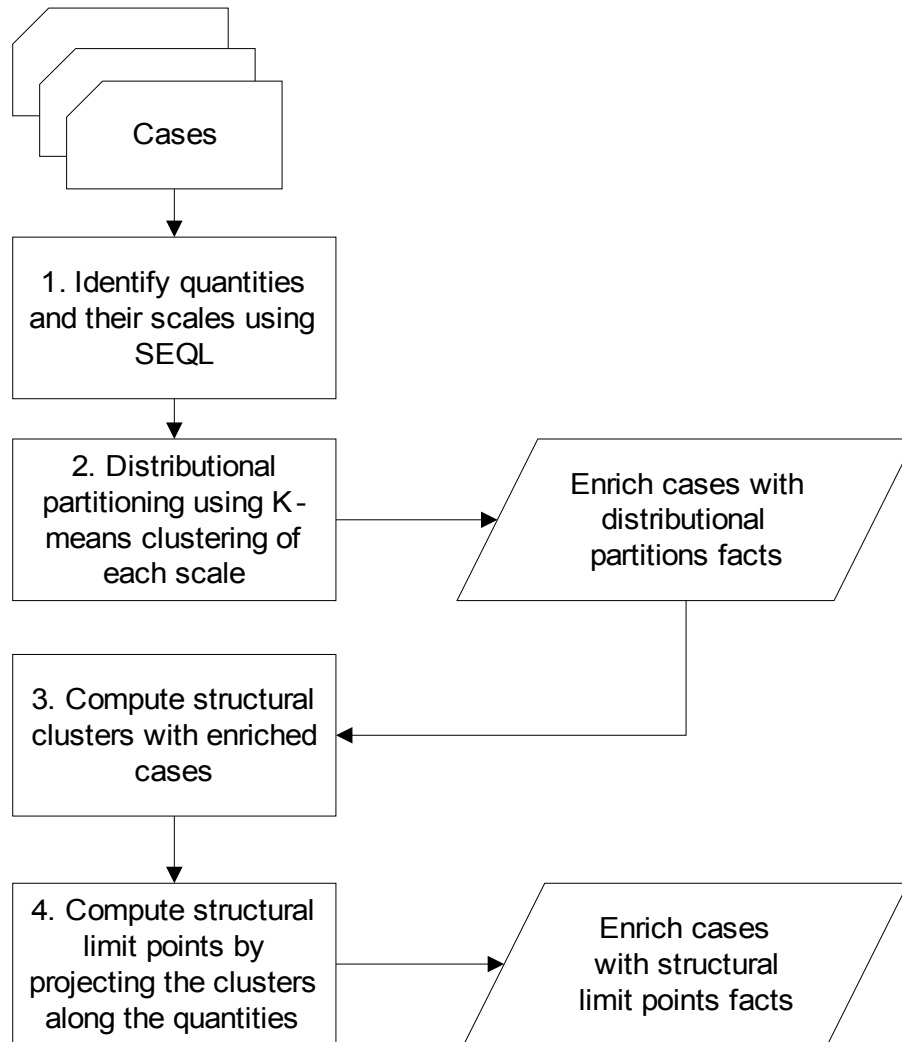


Figure 3.2: A schematic of CARVE

We believe that the key learning process is comparison. Comparison identifies the quantities for which it is meaningful to build symbolizations. Structural alignment of two descriptions provides us with quantities that align. Alignable differences [Gentner and Markman, 1994] accumulated over a large number of examples gives rise to *alignable variance*. The alignable variance is the set of different values that are taken on by an aligned quantity. Consider one starting out to learn

about animals and recording various length dimensions. By comparing various exemplars, one might first recognize some of these dimensions that go together, for example, height and length. This gives rise to a differentiation of the length dimension into two quantities, height and length. Furthermore, the alignable variance accumulated for a quantity is the scale of values for that quantity. Early on in development, similar processes may give rise to the discovery and differentiation of dimensions. We call this process symbolization by comparison. CARVE is a computational model of this process. It provides an account of the generation of both dimensional and structural partitions. The input to CARVE is a set of examples represented as *cases*. Cases are collections of facts in predicate calculus that describe an object or an episode. CARVE generates symbolic, qualitative representations of quantities in the examples, and the output is examples enriched with this symbolic representation. Figure 2 shows the main steps in CARVE. We discuss each of the steps in turn.

3.6.1 Identifying Quantities and Scales

We begin with a set of examples which contain quantitative attributes. A complex example may contain various instances of the same type of quantity, for example, a description of a country may include its population, as well as populations of cities and states belonging to the country. Not all populations should be put on the same scale. It is reasonable to say that New York is large and Sri Lanka is small with respect to population, even though there are roughly the same number of people in both the places. The goal of this step is to find out the quantity values that should be compared on the same scale. Our claim is that those quantities that are comparable give rise to a scale.

Recall that, according to structure-mapping, we draw analogies between two cases by aligning their common structure. Each case's representation contains entities, attributes of entities, and relations. Structure is the connections between elements in the representation. A simple relation between two entities has a small amount of structure, whereas a more complex relation between other relations in the representation has a deeper structure. SME takes as input two cases: a base case and a target case. It finds possible correspondences between entities, attributes, and relations in the two cases. It combines consistent correspondences to produce mappings between the cases. SME attempts to find mappings which maximize *systematicity*, the amount of structural depth in the correspondences. SME also produces candidate inferences about the target by identifying attributes and relations in the base that lack corresponding elements in the target.

SEQL [Skorstad et al, 1988; Kuehne et al, 2000] provides a framework for making generalizations based on computing progressive structural overlaps of multiple exemplars. In its default mode, SEQL works in the following way: when it encounters a new case, it uses SME to compare that case to the known generalizations. If the new case aligns with a sufficient amount of the structure in one of the generalizations, the case is added to that generalization. Any part of the generalization's structure that does not align with the new case is removed, so that the generalization continues to represent only the structure found in all of its exemplars.

The examples are given as an input to SEQL, which builds generalizations out of them. At this step, CARVE extracts the aligned quantities from the generalizations, and the set of values taken by these quantities. The output of this step is quantities and their scales.

3.6.2 Distributional Partitioning

The job of the dimensional partitioning step is to find three partitions, corresponding to Low, Medium and High ranges of the values that the quantity takes. These partitions are currently generated using a k-means clustering algorithm. It is possible to plug in different heuristics that partitions the values into ranges of values. Heuristics based on central tendency and percentiles do not work for Zipf-like distributions as they have very high variance. More empirical data is needed to determine what set of heuristics people use to make these partitions, and when they work. We believe, that depending upon the distribution of data, people will use different partitioning strategies. The clustering scheme used is useful across different kinds of distributions and can be used incrementally without a priori knowledge of distributions. For each fact about the value of a quantity, we then add a `High/Medium/LowContextualizedValueFn` to the case depending upon which range that numeric value fell in.

3.6.3 Structural Partitioning and Projection

The goal of structural partitioning is to find the structural clusters in the cases (for instance, groups of developing and underdeveloped nations) and project these clusters on to various quantity dimensions. The cases produced at the end of the dimensional partitioning step are given as input to SEQL. In figure 3.3, we see the output of SEQL as three generalizations S1, S2 and S3 and some leftover cases that did not fit any of those. Let's consider two quantities `Quantity1` and `Quantity2`. The projection of a cluster on a quantity is the range of values for that quantity in the cluster. For `Quantity1`, we see that the projections from all the three

generalizations overlap. On the other hand, the projections of the generalization on Quantity2 are non-overlapping. We have marked by L_1 and L_2 the boundaries for these ranges. Notice the predictive power of knowing that for a specific case the value of Quantity2 is less than L_1 . We not only know about the quantity value, but about the generalization to which the case belongs, and so can predict other causal properties of it. For instance knowing that a country is a developing country allows us to predict other aspects of it. Structural partitions are a reflection of our deep understanding of the causal and correlational structure of examples. In science, phase transitions, and structural distinctions in socio-economic dimensions were not easily discovered. Finding meaningful structural partitions is dependent upon the richness of such causal and relational knowledge in the example descriptions.

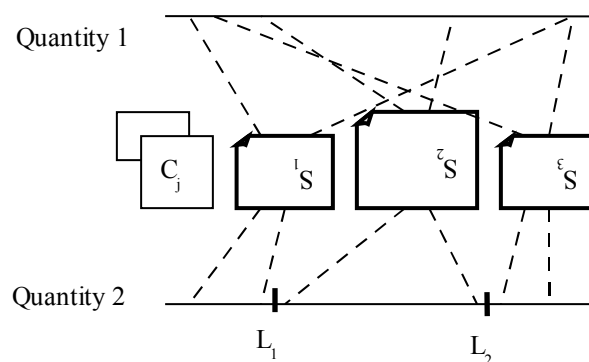


Figure 3.3: Finding structural limit points by projection.

Once the structural limit points are found, structural facts for each quantity are generated as ordinal representations with respect to the structural limit point. These facts are then added to the cases. Next we describe the analogical estimation task, where we show a functional evaluation of the representations generated by CARVE.

3.7 Analogical Estimation

In this section we explore the question: How do people solve quantitative estimation problems, especially in knowledge and experience rich domains? Analogical estimation is using a similar example to make a numeric estimate. For example, consider answering the question: How much does a two bedroom apartment in the Rogers Park neighborhood of Chicago cost? A similar apartment in the same neighborhood or close by might provide a useful estimate. In order to use analogies to make numeric estimates, our analogical matching algorithms should be sensitive to quantities in the first place. We show that the representations generated by CARVE help make more accurate estimates. We begin with a brief and selective survey of relevant research.

A dominant paradigm for research in quantitative estimation has been *anchoring and adjustment* [Tversky and Kahneman, 1974; Kahneman, 1992]. This heuristic says that people make estimates by starting from an initial value that is adjusted to yield the final answer. The insufficient adjustment bias suggests that the adjustment process is biased towards the initial values. This work has been done in domains ranging from information rich, real-world estimation (Northcraft and Neale, 1987) to impoverished guessing (Tversky and Kahneman, 1974, Tenenbaum, in press). One demonstration of the anchoring bias involves the subject making a comparison with an incidental number, called the anchor. Later on, when subjects are asked to come up with a quantitative estimate, then their answers are biased towards the anchor they were initially given. For example, participants were asked to compare the percentage of African nations in the UN as being as higher or lower than an arbitrary number (25% or 65%). Following this, they were asked to estimate the percentage of African nations in the UN. The mean estimates for the subjects who received the high anchor was 45% compared to 25% for the

low anchor. Anchoring effects have been found with both domain experts and novices, e.g., real estate agents and students estimating an appraisal value for a house after touring through it (Northcraft and Neale, 1987).

There is a growing body of evidence (Mussweiler and Strack, 2001; Chapman and Johnson, 1999) indicating that anchoring is not a purely numeric phenomenon, but has semantic underpinnings. Mussweiler and Strack's selective accessibility model of anchoring suggests that the anchor causes increased accessibility of anchor-consistent knowledge. For example, with the high (65%) anchor in the Africa example, facts like "Africa is a large continent" and "There are more African countries than I keep in mind" are retrieved. The final numeric estimate is generated based on the easily accessible knowledge (Higgins, 1996), so their estimate is heavily influenced by anchor-consistent knowledge. This line of argument proposes a semantic priming based explanation of the anchoring and adjustment phenomena. Epley and Gilovich (2005) argue that the standard "experimenter-provided" anchors behave differently from "self-generated anchors," and the former are not very informative about the actual process of adjustment.

Brown and Siegler (1993) explored quantitative estimation in a three-phase experimental paradigm. First, participants are presented a set of items and asked to estimate the value of a particular quantity for each item (e.g., populations of countries). Next, they learn the actual value of a subset of items (called seed items). Finally, the subjects re-estimate the values of items in the initial set. The experimenters found improved estimation as a result of seeding (Brown and Siegler 1993, 1996). They found that people access two independent sources of knowledge while generating estimates: 1) Metric knowledge: information about the numerical properties of the quantity, and 2) Mapping knowledge: non-numerical information about the domain which could

be used to order items relative to one another with respect to that quantity. Brown and Siegler (2001) have shown that seeds behave differently than anchors. Seeds provide both metric and mapping knowledge by providing feedback (“small European countries have fewer people than I would have guessed”). In contrast to anchors, seeds can push estimates of target items away from their own values. These data suggest that quantitative estimation is not a purely numeric task: non-numeric knowledge is used to construct estimates

3.7.1 Verbal Protocols of Analogical Estimation

To observe how experts utilize similarity and causal relationships in real world estimation tasks, we conducted a protocol analysis of experts doing realistic estimation tasks. The goal of this study is to determine the extent to which experts use analogical estimation, specifically if they use analogical anchors and make causal adjustments while estimating. We interviewed two experts in two different domains, an employee at a used car dealership with two years of experience and a apartment realtor with five years of experience. We constructed numerical estimation questions by taking items off of public listings within areas of their expertise, online advertisements for housing rentals and used cars, and removing the asking price.

After performing some warm up exercises as recommended by Ericsson and Simon (1993) to increase the participants willingness to reason aloud, we would present each problem. The subjects were given as much time as they wanted to answer each question, and they were occasionally prompted with questions such as “What are you thinking about right now?” if they remained quiet too long. Each question was a complete listing of a car or an apartment with the

price removed. Apartment listings were from the Craigslist²² website containing all details such as size, location, utilities, etc. This is the same information an apartment seeker would have access to in their preliminary search. Car listings were taken from a Carmax²³, a popular used car website. Each description contained a picture of the specific car and a standard format containing relevant information about the car. After each estimate the participants were asked to explain their answer. The apartment trial consisted of eight questions, while the used car trial consisted of seven questions.

We coded the protocols for three aspects of analogical estimation:

1. *Analogical reminders*: Explicit references to remembered prototypes of a class, or specific instances that were similar to the problem, e.g.,
 - a. “This [Lakeview apartment] would go for \$700-750 in Rogers Park”
 - b. “These [cars] are just shy of \$30,000 brand new.”
2. *Causal adjustments*: Explicit references to other causal quantities during the estimation process, e.g.,
 - a. “You know [parking spaces] are worth more in Lakeview”
 - b. “These [cars] are particularly hot right now because of higher gas prices.”
3. *Non-alignable features*: Explicit adjustments based on features present in one of the cases, e.g.
 - a. “If [the Cadillac Escalade] is black it is 1,000 dollars more”
 - b. “I’m going to raise [the estimate] a little, I was not thinking about the deck.”

²² <http://chicago.craigslist.org/>

²³ <http://www.carmax.com/>

Table 3.1 below summarizes the data collected in protocols. It indicates that analogical estimation is a common strategy used by experts solving estimation tasks.

Analogical Estimation Aspect	Cars (n=7)	Apartments (n=8)
Analogical reminders	7	11
Causal adjustments	11	7
Non-alignable adjustments	5	12

Table 3.1: Number of analogical estimation occurrences.

3.7.2 A Theory of Analogical Estimation

In this section, we describe our theory of analogical estimation. Analogical estimation is a specific kind of analogical inference, namely, inferring the quantitative value of an unknown based on a known value from a similar example. For example, when trying to estimate the rent for an apartment, one might retrieve from memory a similar apartment in the same neighborhood. The value from the analogical reminding serves as an *analogical anchor*. As a first pass, this analogical anchor is evaluated for its plausibility for the value sought. Analysis of the comparison between the problem and the reminding provides the grist for computing adjustments from the anchor to improve the estimate: for example, one might notice that the apartment that they were reminded of is smaller, and is in a slightly less desirable location. In this example, there are two causal assumptions about apartment rents:

1. A larger apartment has higher rent, all things being equal.
2. The more desirable the location, the higher is the rent, all things being equal.

Note that these are qualitative models and the relationship described above are qualitative proportionalities [Forbus, 1984]. These facts suggest that the estimate of rent should be more

than the rent of the reminded apartment. Just how much more? The effect of location on rent can vary, and in some neighborhoods, it might be stronger than others. At this point, one can use other examples to determine just how strong that effect is. We call these adjustments based on causal knowledge *causal adjustments*. The final estimate is generated by adjusting the analogical anchor to reflect the causal adjustments.

3.7.2.1 Analogical Anchors

Analogical estimation begins with searching and retrieving from memory other examples that are similar in ways to warrant being plausible estimates for the quantity sought. The reminders retrieved could be specific exemplars, or generalizations (Kuehne et al, 2000). The value of the quantity sought in the reminding is an analogical anchor. Analogical anchors are similar to self-generated anchors (Epley and Gilovich, 2004) in the sense that they are generated by the subject spontaneously as they solve the estimation problem. An example of a self-generated anchor is the freezing point of water while estimating the freezing point of vodka. However, there are two important differences between self-generated and analogical anchors: 1) the specific stimuli used in studies on self-generated anchors were designed to activate one strong anchor across subjects, and 2) self-generated anchors could be salient points on the dimension, irrespective of their relevance to the current problem. When individuals' knowledge of the domain of estimation is sparse, they will recruit any salient points on the dimension to guide their estimation. Most of the self-generated anchors fall in this category. However, with more experience in the domain, one might have access to a number of similar situations, possibly richly represented with causal knowledge and relationships between quantities. These are analogical anchors.

The similarity between two objects doesn't necessarily warrant the inference that values of all the quantities for two objects are similar. For example, two similar basketball players might have similar height, but not necessarily two professors. This notion of what features can be inferred from a similar example was called *projectability* by Goodman (1955/1983). Projectability is based on *centrality* of the feature (Hadjichristidis et al, 2004). A feature is central to the extent that other features depend upon it. In the above example, height is central to basketball players, but not to professors. We have operationalized this notion of centrality as the structural support (Forbus et al, 1997) of the inference in computation of similarity using the SME.

3.7.2.2 Causal Adjustments

A key component of expertise is an understanding of the underlying causal structure of the domain. An important type of causal relationships is qualitative proportionalities (Forbus, 1984). Qualitative proportionalities indicate a monotonic relationship between two variables. These are useful for numeric estimation as they provide the ordinal direction for adjustment, e.g., a larger apartment has a higher rent, all else being equal. In verbal protocols presented in the section 3.8, we find that people commonly refer to such qualitative proportionalities while estimating. Such adjustment based on qualitative proportionalities are called causal adjustments.

However, it is not at all clear how to figure how much to adjust, as the qualitative proportionality only indicates a monotonic functional relationship between two variables, and does not tell us anything about the strength of this relationship. Let's suppose that the estimation problem involves two quantities: x and y , and that the unknown quantity we are trying to

estimate is y . Further, we are given that there is a positive qualitative proportionality between x and y , i.e.,

$$y = qprop+(x)$$

where $qprop+$ indicates a monotonically increasing function. Suppose we were reminded of a similar situation, where the qualitative proportionality also holds true, and value of both quantities, x^* and y^* are known. Based on this, we can conclude if y will be more or less than y^* , as a result of the monotonic dependence.

$$sign(x-x^*) = sign(y-y^*)$$

At this point, we cannot conclude anything about how much more or less y is than y^* without making assumptions about the nature of the function $qprop+$. However, if we know a few more examples where this qualitative proportionality is valid, i.e., data points on this function, we can use that to approximate the dependence by fitting a curve over those points. Let's assume we can recall a small set of situations where this proportionality is valid, $\{(x_i, y_i)\}$. Based on these, we can obtain an approximate estimate of the dependence between y and x ,

$$y = Q^*(x)$$

The suggested adjustment based on this approximation is,

$$adjustment = Q^*(x) - y^*$$

So, the causal adjustment is obtained by using an approximate estimate of the functional dependence between the quantities. The error in causal adjustment then is the discrepancy between this estimated dependence and that exists in the world. So, if one falsely believes that there is a strong relationship between two variables, then one is likely to produce a causal

adjustment that is higher than needed. As opposed to the insufficiency results for adjustment, we expect errors in causal adjustments to be based on people's understanding of qualitative proportionalities in the world, and thus causal adjustments need not be insufficient. There is evidence to support that people can and do estimate correlations with very few samples, on the order of five (Kareev, 1997). We would expect causal adjustment to be affected by systematic biases in detecting correlations.

3.7.2.3 Adjustment based upon non-alignable features

Comparison between the reminding and the problem might reveal features that are present in one but do not have a corresponding feature in the other (Markman and Gentner, 1997). For example, one might retrieve a similar apartment, but one whose rent includes parking space. This is a sub-problem of the original estimation problem that is solved independently using the same mechanisms, e.g., one will invoke analogical estimation for the parking space.

3.7.3 KNACK: A Computational Model of Analogical Estimation

In this section, we present Knack, a computational model of the theory of analogical estimation presented in the previous section. Figure 3.4 shows a high level description of Knack's algorithm. Knack's experience consists of a case library, a set of examples. An estimation problem is presented to Knack as a case, a set of predicate calculus expressions that represent all the information in the problem. Knack retrieves a few examples from the case library that are most similar to the problem at hand. The best reminding is used to generate the analogical

anchor. The projectability of this inference is determined by looking at the structural support returned by SME. At this point, we extract all the aligned causal relationships that involve the quantity sought. A linear regression is performed for all the retrieved data points for each causal relationship. This gives us an approximate sense of the strength of the causal relationship. We compute adjustments for each causal relationship based on this approximate fit generated by linear regression. If the fit violates the expected qualitative relationship, then the adjustment suggested by this relationship is ignored. All valid causal adjustments are added to the analogical anchor to generate the estimate.

1. Retrieve similar examples (n=5) from memory
2. Select the most similar example's value as the anchor
3. Check if this is a plausible anchor by computing projectability
4. Find all causally connected quantities from the common causal structure in the retrieved examples
5. For each causally connected quantity
 - 5a. Compute adjustment via linear fit with the retrieved examples
 - 5b. Check adjustment with expected directionality of causal relationship
6. Apply all applicable adjustments to the anchor to generate the estimate

Figure3.4. The KNACK algorithm

3.7.4 Estimating Basketball Statistics

To illustrate the above ideas, we report results from an experiment in the domain of estimating basketball player statistics (e.g., Points per game, Assists per game, height, etc.). This domain was chosen because there is a host of numeric information easily available, and there are interesting causal relationships between quantities, e.g., being tall helps to rebound. We selected thirty players such that they were reasonably different, six from each of the five positions on the court. We built a case library in which each basketball player was represented as a case. The

average case had twelve facts, including four qualitative proportionalities, e.g., minutes per game is qualitatively proportional to points per game. We compared Knack to baseline analogy by running two trials. The baseline trial makes estimates by choosing the value for the dimension on the player selected by MAC/FAC as the best reminding. The Knack trial utilized CARVE to enrich the cases with symbolic representations for the quantitative facts. On an average, this added ten facts to every case. For example, CARVE generates the following qualitative representation for each quantitative fact:

Quantitative fact	Qualitative representation
(seasonThreePointsPercent JasonKidd 0.404)	(isa JasonKidd (HighValueContextualizedFn seasonThreePointsPercent BasketballPlayers))

In both trials, the facts mentioning the sought after dimension were filtered out of the question case. The trials were conducted in round robin format in which estimates were recorded for every player and every dimension.

We present the comparison of error in estimates generated using baseline analogy and Knack. Knack's estimate are significantly more accurate ($p < 0.05$) for four out of six dimensions across all players. Although the error for assists per game appears to be higher for Knack, the difference is not significant. Similarly, there is no significant difference in errors for free throw percentage. The free throw percentage dimension was not causally related to any other quantities, and the assists per game is highly variable across our dataset. This is because our representation implies that these dimensions are not causally central.

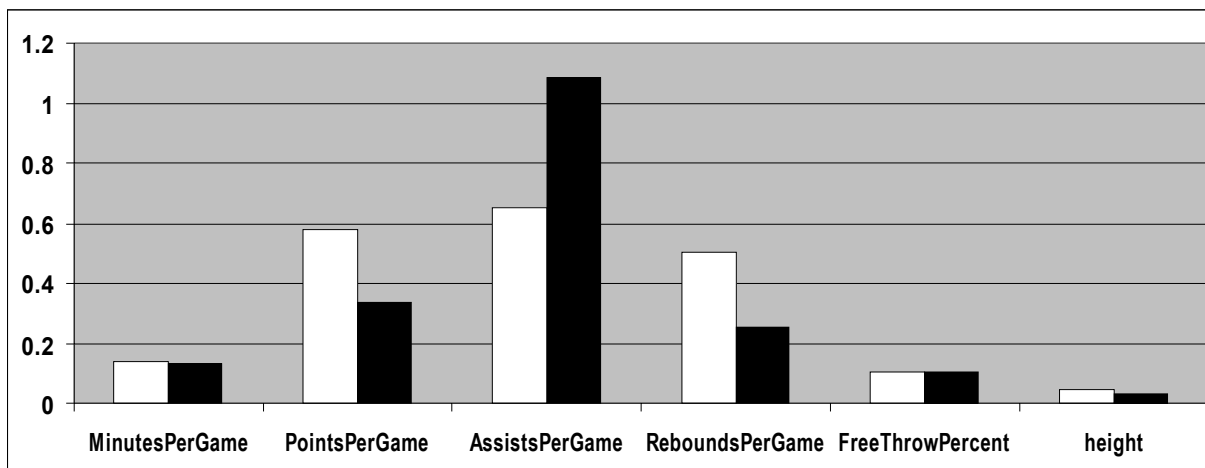


Figure 3.5: Comparison between normalized mean error, $ABS(estimate - value)/value$ of estimates by dimension. White is baseline error and black is Knack's error.

When looking at the amount and direction of the adjustment, we consider no adjustment to be an under adjustment. Qualitative proportionalities are directional, therefore Knack only made causal adjustments for points per game, rebounds per game, and assists per game. Knack handles the contradiction between a computed adjustment direction and the sign of the qualitative proportionality by ignoring it. This leads to a systematic under adjustment for these dimensions.

Knack demonstrates how similar examples can be used to find analogical anchors in quantitative estimation tasks. These analogical anchors are similar to the self-generated anchors studied by Epley and Gilovich (2005). They found that with forewarning and incentives subjects could overcome the insufficient adjustment bias. The Knack model hypothesizes that under adjustment is more likely when the subject is less confident in the nature of the adjustment. One way in which Knack could model the increased effort in overcoming the bias would be when faced with contradictory adjustment directions, to retrieve more and more examples until the qualitative proportionality was satisfied. Knack is consistent with Mussweiler and Strack's

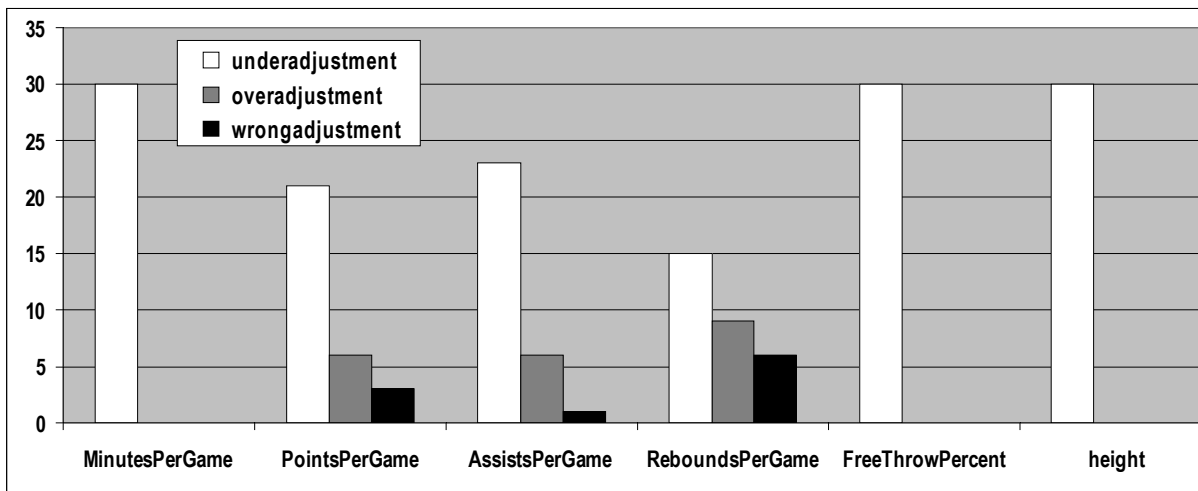


Figure 3.6: Comparing under adjustments, over adjustments, and wrong adjustments made by Knack over all the estimation problems.

(2001) claim that anchors cause subjects to activate anchor consistent knowledge. The symbolic encoding of the anchor as a plausible answer will bias the retrieval of examples that are consistent with it. We expect that anchoring and adjustment will show a strong effect of experience with other examples in the domain: 1) the most similar examples will provide the anchors, and 2) adjustment need not be insufficient, but will mirror the strength of the causal relationships that follow from the subject's experience. More psychological experiments have to be done to verify these predictions.

3.8 Conclusions

We presented the symbolization by comparison theory of quantity sense that claims that we learn about quantities by abstracting qualitative representations via the process of comparison. We presented CARVE, a computational model that implements this theory. We evaluated CARVE in

the context of an analogical estimation task, and showed that representations generated by CARVE increase the accuracy of estimates.

Chapter 4: System Description and Examples

Problem solving is a process that takes us from a problem to its solution. This path consists of finding and applying relevant domain knowledge and using strategic and heuristic knowledge to guide the process. To test our theories of quantity and back of the envelope reasoning described in the previous two chapters, we have implemented *BotE-Solver*, a system that generates back of the envelope estimates in multiple domains. We evaluate BotE-Solver on the practice problems on the Fermi problem section from the Science Olympics²⁴. This set consists of thirty five problems, some of which are shown in Figure 4.1. The system can solve all of these problems.

In this chapter, we describe the design and implementation of BotE-Solver. We present examples that highlight various aspects of its operation and heuristic knowledge. We first describe the general problem-solving features of the system, and then focus on implementation of the heuristic methods described in Chapter 2.

²⁴ <http://www.physics.uwo.ca/olympics/>

If the mass of one teaspoon of water could be converted entirely into energy in the form of heat, what volume of water, initially at room temperature, could it bring to a boil?

How much energy does a horse consume in its lifetime?

How many bricks are there in London?

How many electrons could a fully charged 12 volt car battery release before it was completely discharged?

What is the mass of all the automobiles scrapped in North America this month?

Figure 4.1: Some examples from the Science Olympics corpus

4.1 BotE-Solver

A computational model of problem solving provides a representation for problems, access and retrieval mechanisms for relevant domain knowledge. It also needs to have access to heuristic knowledge, which is used when the problem is complex and the answer is not directly found. It needs to maintain the workspace, where it keeps track of its progress made during problem solving. Additionally, it needs to have learning mechanisms that allow it to improve from experience. Figure 4.2 shows a high-level architecture of BotE-Solver. The components are divided into four categories:

1. *Learning*: BotE-Solver learns qualitative representations of quantity using the CARVE system described in the previous chapter. These representations help it to retrieve better examples for analogical estimation. This learning is “offline,” in the sense is done not in response to a given problem, but when examples are accumulated, in batch. Currently

CARVE-generated representations are used for estimation in the basketball domain only, which was described in Chapter 3.

2. *Knowledge*: BotE-Solver’s knowledge base (KB) consists of a 1.2 million fact subset of Cycorp’s ResearchCyc²⁵ KB plus knowledge represented and developed in our research group. This knowledge base contains ground facts, axioms, case libraries consisting of episodes to be used for analogical estimation, and heuristic methods.

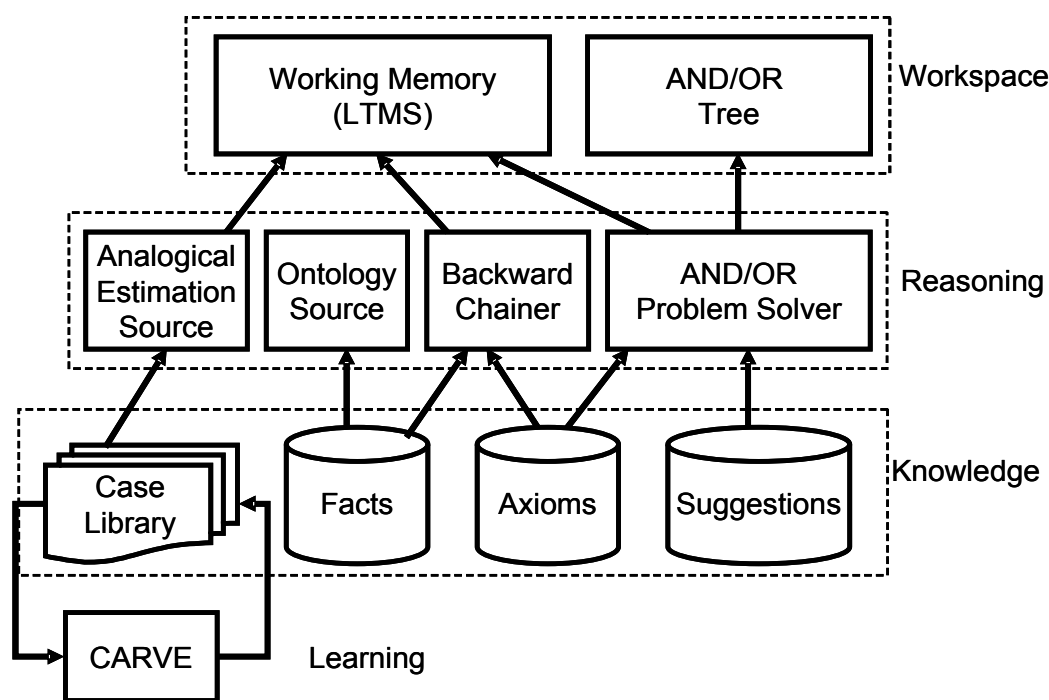


Figure 4.2: Architecture of BotE-Solver

3. *Reasoning*: BotE-Solver generates an estimate via lookup, analogical estimation, backward chaining, in that order. If those fail, an AND/OR-tree based problem solver is used to apply heuristic methods to transform it into other problems. These are the

²⁵ All references to Cyc in this chapter refer to this specific version of ResearchCyc.

heuristic methods described in Chapter 2. The heuristic methods are implemented as suggestions [Forbus and deKleer, 1993] which is described in more detail in Section 1.2.

4. *Workspace*: The AND/OR tree keeps track of the problem decomposition and work done on the problem. Furthermore, a Logic-based Truth Maintenance System [Forbus and deKleer, 1993] is used to implement the working memory. This records dependencies and caches facts that have been retrieved from the KB or inferred via reasoning.

The next sections describe these components in more detail.

4.1.1 Knowledge and Reasoning Infrastructure

The Knowledge Base (KB) and the FIRE reasoning engine are part of background infrastructure that this work builds on. Problems, solutions, strategies are all represented uniformly and stored in this KB.

FIRE is a federated reasoning architecture built by our group at Northwestern University in collaboration with PARC, Inc. FIRE is described in detail elsewhere [Forbus et al., 2006] and here we will describe only the relevant components briefly. FIRE uses a database for storing the knowledge base which facilitates scaling up, persistent storage, and portability with regards to knowledge. It provides support for analogical reasoning using Structure-Mapping Engine and MAC/FAC [Forbus et al, 2002]. It provides the conventional ASK and TELL interface to the knowledge base, and QUERY which uses backward chaining. The federated aspect of its architecture is in the fact that it allows *reasoning sources* as a mechanism for procedural attachment for doing specialized reasoning, such as spatial reasoning [Forbus et al, 2003]. A

reasoning source provides procedural attachments that handle specific queries, based on the predicate involved and the query pattern.

The backward chaining is partitioned, i.e., rather than chain through all the axioms in the knowledge base, it is limited to axioms contained in explicitly loaded chainers, subsets of the knowledge base that have been identified as potentially relevant for some class of queries. The next level of reasoning is SOLVE which uses *suggestions* to solve problems by producing AND/OR decompositions of the problem. Before we describe SOLVE, we describe suggestions as means for encoding heuristic knowledge.

4.1.2 Implementing Heuristic Methods Via Suggestions

Suggestions are declarative fragments of knowledge that specify problem solving strategies and plans. A suggestion provides a decomposition for the problem. Figure 4.2

```
(defSuggestion SphericalVolume
  (valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)
  :documentation "Consider a spherical cow for computing volume"
  :test (and (isa ?object ThreeDimensionalThing)
             (uninferredSentence (shapeOfObject ?object ?shape)))
  :subgoals ((valueOf ((QPQuantityFn extensionParametersOfObject)
                      ?object) ?size))
  :result-step (evaluate ?volume (TimesFn 12
                                         (ExponentFn
                                          (QuotientFn ?size 2) 3))))
```

Figure 4.3. An example suggestion

shows a suggestion `SphericalVolume` that describes an approximation for computing volume. The trigger `(valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)` specifies that this suggestion applies to problems which are seeking bindings for

volume of an object. The test condition checks if we know that the object in question is a three dimensional object, and confirms that we do not know the shape of this object. It triggers a subgoal to estimate any linear extent parameters. The result step then computes the volume of the object as if it was a sphere with that extent as diameter. In general, there are four parts to a suggestion:

1. *Trigger*: The form which is query for which the suggestion might be applicable.
2. *Test*: Additional test conditions which must be true in order for the suggestion to work.
3. *Subgoals*: A list of forms that this suggestion decomposes the current problem into. These are AND-subgoals, meaning if any one of them fails, this suggestion fails to solve the original problem. These subgoals are fully ordered.
4. *Result-step*: The final step of the suggestion, which combines the answers to the subgoals. This form often uses `evaluate`, which is a mechanism to implement predicates that require computation, e.g., arithmetic functions.

```

((ist-Information BotESuggestionsMt
  (chainerContains (ChainerFn BotESuggestionsMt)
    (<==
      (suggestFor (valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)
        SphericalVolume)
      (isa ?object ThreeDimensionalThing)
      (uninferredSentence (shapeOfObject ?object ?shape))))))
(ist-Information BotESuggestionsMt
  (comment SphericalVolume "Consider a spherical cow for computing volume"))
(ist-Information BotESuggestionsMt
  (suggestionResultStep SphericalVolume
    (evaluate ?volume (TimesFn 12 (ExponentFn (QuotientFn ?size 2) 3))))))
(ist-Information BotESuggestionsMt
  (suggestionSubgoals SphericalVolume
    (TheList
      (valueOf ((QPQuantityFn extensionParametersOfObject) ?object) ?size))))
(ist-Information BotESuggestionsMt
  (suggestionGoalForm SphericalVolume
    (valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)))
(ist-Information BotESuggestionsMt (isa SphericalVolume Suggestion))

```

Figure 4.4: Predicate calculus statements generated by the SphericalVolume suggestion

Inside the suggestion, the variables are ordered, such that any variable introduced can be used in the subsequent parts of the suggestion. The `defSuggestion` macro mentioned above is a facility for the suggestion author. It expands the suggestion into predicate calculus statements that are stored in the KB. Figure 4.4 shows the predicate calculus statements corresponding to the suggestion above. The library of suggestions used by BotE-Solver is included in Appendix B.

4.1.3 Tracking problem solving progress by AND/OR trees

BotE-Solver uses AND/OR trees²⁶ to track the progress as it is working on a problem. AND/OR problem solving is not novel, and our implementation is based on Slagle (1963) and Nilsson

²⁶ Because the solutions are obtained and cached in a TMS, we get the functionality of an AND/OR graph, i.e., we don't re-solve an already solved node, although the underlying representation is a tree. We want to be able to incrementally generate solutions; sharing nodes as in a graph does not work when the nodes are generators of solutions.

(1994). We made some interesting design choices with regards to incremental computation of solutions. Keeping track of all solutions can be computationally expensive, especially for a problem solver using a large KB, as there can be prohibitively large number of combinatorial possibilities. In this section we briefly describe our AND/OR solver.

An AND/OR tree is a convenient representation for problem decomposition and allows for reuse of work done during problem solving as well as provides the grist for providing explanations to the solutions generated. The mapping between the AND/OR tree and our representations is very direct. For a problem, there could be many applicable strategies, any one of which succeeding lead to a solution to the problem. This results in an OR node in the tree. A suggestion, on the other hand, introduces one or more subgoals all of which have to solved in order to solve the original goal. This results in an AND node in the tree. An AND/OR decomposition lets us keep track of dependencies between the original problem and new subgoals introduced. During the course of problem solving, a node can be:

- 1) SOLVED: An OR-node is solved when any one of its children gets solved, and an AND-node is solved when all of its children are solved.
- 2) FAILED: An OR-node fails when all of its children fail, and an AND-node when any one of its children fail.
- 3) MOOT: A node is moot when it is not solved or failed, but when there is no point on working on it at the current point. So, if any one of the siblings of OR-nodes has succeeded, the other siblings are MOOT-VIA-SUCCESS, as at any point we are interested in finding one solution, so if one strategy has succeeded, we don't want to pursue others right now. However, we might come back and un-moot the other strategies

if at some point we want more solutions, or after propagating this solution upwards in the tree we find that the original goal is still not solved. A strategy can generate many solutions, and we try the next sibling strategy only if we have exhausted all the solutions that this strategy has to offer. On the other hand, if an AND-node fails, its siblings are MOOT-VIA-FAILURE as there is no point in working on them, as the parent suggestion has failed as a result of one of its children failing. However, if later we find that we can solve the subgoal that failed by working more, we can un-moot the siblings.

These inferences are made by maintaining flags at each node, which are updated and propagated after every unit of problem solving.

As BotE-Solver works on a problem, it maintains its progress in an AND/OR tree as mentioned above. It also maintains an agenda, which is a list of things that it can do next. The agenda consists of suggestions that have been found that it can try, and subgoals that have been suggested. The agenda is ordered by difficulty estimates, such that the first thing on the agenda is the easiest one. BotE-Solver starts with enqueueing the original goal on to the agenda and running the main loop. The rest of the discussion will explain what happens at some point in midst of problem solving when we have done some work and have an already expanded AND/OR tree.

There are two different ways in which new solutions are generated in solve:

1. *AGENDA processing*: The original goal hasn't been solved yet, and we are either trying to find suggestions that will solve it, or working on the subgoals that were suggested. It picks the easiest thing off the agenda. If it is a goal node, then sees if it can be solved by a primitive operation. If that fails, it gathers applicable suggestions via backward chaining. Found suggestions are added as the children of the original goal and enqueued on the

agenda. If it is a suggestion node, then it instantiates the first subgoal of the suggestion node as a child node of the suggestion in the graph, and enqueues it on the agenda.

2. *IN-PLAY processing*: This happens when the original goal has been expanded into a graph all of whose leaf nodes are solved. Now, no more problem solving needs to be done, and we can keep generating new solutions until we have exhausted all possible bindings found at the leaf nodes. We call a node that is solved and can possibly generate more solutions as an IN-PLAY node. Every subgoal maintains a pointer to the current IN-PLAY suggestion. The main concern of IN-PLAY processing main is to properly update what bindings have been already used.

All the bindings that are found as a result of a successful solution are maintained at the nodes locally and only those that are of interest to the parent from the first successful combination of the bindings are propagated upwards. Each node maintains a marker to indicate the bindings that it has already used, and these are updated to make sure we exhaustively go through the space of combination of bindings from the subgoals. The combinatorial possibilities of bindings from subgoals can be large. For example, consider a suggestion whose three subgoals are solved by a primitive operation, corresponding to leaf nodes in the tree. For these leaf nodes if we found 20, 50 and 50 successful bindings, we have fifty thousand combinations of bindings that could possibly lead to the parent. Each of these combinations is tried one by one until a solution for the parent is found.

4.2 Evaluating BotE-Solver

Evaluating a system like BotE-Solver is hard, as BotE reasoning spans a wide spectrum of commonsense, scientific, and policy-making domains. This makes it hard to generate an exhaustive or representative set of BotE problems.

The corpus of problems we chose to use comes from the Science Olympics organized by the University of Western Ontario for high school students. Science Olympics is a generic name for a set of science and mathematics related competitions at all grade levels. The Science Olympics for high school students (Division “C” in the US) has typically about twenty different events like “Ecology,” “Forensics,” “Robot Rambler,” “Sound of Music,” and “Fermi Questions.” Each of events tests different types of scientific, mathematical and experimental skills and teamwork. Teams of five or so participate and nationally compete. The set of events changes every year, however, the Fermi Questions section has remained in all variants of Science Olympics competitions. An order of magnitude answer is the goal in this competition and partial points are also awarded. The scoring is as follows: 5 points for the correct exponent, 3 points for the correct exponent ± 1 , and 1 point for the correct exponent ± 2 .

According to Abrams (2005), who has been supervising the Fermi Questions event for over twenty years, the section consists of about 30 questions and scoring 90 out of a possible 150 points will get the team a medal. Only a few out of hundreds of participating teams (schools) achieve this level of performance.

We evaluate BotE-Solver on a set of 35 practice problems from the Science Olympics website from University of Western Ontario. Some example questions are shown in Figure 4.1 at the beginning of this chapter. The full list of problems and BotE-Solver's answers is in Section 4.

Note that for some of the problems, we do not know the “correct” answer. One such problem is “How many bricks are there in London?” We requested the organizers of the Science Olympics for official answers, but they did not respond.

Lets look at an example, the first question in Figure 4.1. BotE-Solver expects a question in predicate calculus, so the first step is to encode the question in CycL.

If the mass of one teaspoon of water could be converted entirely into energy in the form of heat, what volume of water, initially at room temperature, could it bring to a boil?

This question is more conveniently represented as a two part question – 1) What is the heat energy produced by entirely converting one teaspoon of water into energy? And 2) How much water will this amount of heat energy bring to a boil? Lets look at the first question. We introduce a reified event, `TeaspoonIntoEnergyEvent1`, that describes the hypothetical event of converting a teaspoonful of water into energy. Reification allows us to name a complex situation (like an event) and elaborate it using various roles that can be defined that are applicable in that situation [Davidson, 1967].

```
(isa TeaspoonIntoEnergyEvent1 TotalEnergyConversionProcess)
(genls TotalEnergyConversionProcess EnergyConversionProcess)
(objectActedOn TeaspoonIntoEnergyEvent1 Water1)
```

Given that, the question can then be asked as:

```
(valueOf ((QPQuantityFn energyProduced)
          TeaspoonIntoEnergyEvent1) ?energy)
```

This representation of quantity is borrowed from the Qualitative Process Theory ontology [Forbus 1984]. `QPQuantityFn` is a unary function whose first argument is a continuous quantity, and the result of its application is a variable-arity function. The result of applying the

resulting function is a continuous quantity. `valueOf` simply relates the continuous quantity to its numerical value.

It is important to note that there was additional knowledge that was provided to the system for capturing this question. Cyc did not have a notion of `TotalEnergyConversionProcess` and the associated law of $E = mc^2$. It is impossible to answer this question without this knowledge. Being able to answer questions successfully relies on three types of knowledge: 1) *domain knowledge*, 2) *heuristic knowledge*, and 3) *heuristic applicability knowledge*. Domain knowledge is knowledge about objects, relationships and quantities in a domain, e.g., knowing about Energy, Mass and $E = mc^2$. As we work with a KB in progress, it is inevitable to add domain knowledge, as one grounds out the problem solving by knowing such facts.

Heuristic knowledge refers to heuristic methods, e.g., knowing that mass of an object can be estimated by adding up mass of its parts. This is the part where BotE-Solver claims completeness. We claim that the seven heuristic methods used by BotE-Solver are all we need for BotE reasoning. As we encoded the Science Olympics corpus, we did not need to add any new heuristic methods.

Heuristic applicability knowledge refers to knowledge that enables the application of the heuristic, for example, figuring out subparts of an object depends to some extent on the domain. For example, the strategy of adding up subparts requires finding all the contained objects when dealing with tangible objects with physical extent. However, if the dimension in question were time, it requires finding all the sub-events. If the heuristic methods are represented abstractly, but in different domains, the same abstract concepts (like part-whole-ness) might be represented

differently, with different predicates. The heuristic applicability knowledge makes that between general and domain specific knowledge. This type of knowledge needs to be added once per domain of applicability.

4.3 Heuristic Methods in Bote-Solver

In this section, we look at each of the seven heuristic methods used by BotE-Solver. We present example suggestions for them and discuss the heuristic applicability knowledge. This discussion is grounded with respect to Cyc, however, most of these ideas apply broadly.

4.3.1 Ontology Heuristic Method

The ontology heuristic suggests finding the closest point in the ontological lattice that can sanction a guess for the quantity in question. For instance, one can estimate the height of Jason Kidd by noticing that he is a `BasketballPlayer`. However, besides being a `BasketballPlayer`, Jason Kidd is also a `FamousPerson`, `MaleHuman` and `Individual` among others. How can we determine that `BasketballPlayer` is a better collection than the others mentioned for making an inference about height?

There are two heuristics that come from psychological investigation of category structure:

1. *Level of categorization:* Rosch [1978] identifies three levels of categorization: subordinate, basic-level, and super-ordinate. According to this characterization, predictive power of a cue decreases as one goes higher than the basic-level. This rules out collections that are very general like `Individual`.

2. *Projectability*: Even categories at the same level might sanction inferences about certain features and not so about others. For instance, `BasketballPlayer` will provide more accurate estimate about Jason Kidd than `Professor` would about a professor. There is increasing psychological evidence that projectability is based on *centrality* of the feature [Ahn et al., 2000; Hadjichristidis et al., 2004]. A feature is central to the extent that features depend on it. Height is more central to basketball players than professors.

These two psychological heuristics suggest picking something from the ontological lattice as long as it is not more abstract than the basic-level, and the quantity being estimated is central to that class of objects. According to Rosch, the basic-level of categorization is that level of categorization that maximizes the total cue validity of a category. The cue validity is the probability of a given cue x successfully predicting of a given category y . The total cue validity of a category is defined as the summation of cue validities for that category of each of the attributes of that category. In principle, we can compute this in Cyc: however, it is expensive to compute and maintain as new information comes along²⁷. Figure 5 shows the way ontology heuristic method is implemented in BotE-Solver. First, we exclude collections like `Individual` from being used. This is done by automatically excluding any collection that has more than a threshold number of instances. This threshold can be adjusted, it is currently set to a thousand. We also have a minimum threshold on how many instances must exist for the inference to be useful, currently set at 3. We then find a class whose instances have values that have a variance below a threshold. This results, in our Jason Kidd example, preferring

²⁷ Automating the computation of basic-level categories and looking at the discrepancies between human and the basic-level categories predicted by the KB is an interesting way to analyze the plausibility of the ontology and knowledge contained in it. This thesis does not explore this any further.

BasketballPlayer for making a height estimate rather than FamousHuman. In this case, when we have enough instances, we return the arithmetic mean of the values as an estimate.

More often, there are not too many concrete instances to employ the variance heuristic above. In such cases, there is often general information about a whole class of things expressed by `relationAllInstance` predicate, for example –

```
(relationAllInstance massOfObject HomoSapiens (Kilogram 2 400))
```

This statement says that for every instance of `HomoSapiens`, it is true that their mass is between 2 kilogram to 400 kilogram. In this case, we return the geometric mean of this range as an estimate. We use the geometric mean here, as the `relationAllInstance` predicate usually expresses the total range which might include outliers. The geometric mean provides a more stable estimate in such a case.

1. Given a problem $(Q \ O \ ?V)$, for each class O' such that $(isa \ O \ O')$ or $(genls \ O \ O')$ and is not a member of `NotForOntologyHeuristic`
 - If there are known instances $\{I\}$ of O'
 - Calculate number of $\{I\}$
 - Calculate the standard deviation $?V_i$ such that $(Q \ I \ ?V_i)$
2. Return the arithmetic mean of $?V''$ from the O' for which standard-deviation $\{?V''\}$ is minimum.
3. If no estimate found, then, for each O' such that $(isa \ O \ O')$ or $(genls \ O \ O')$
 - If range of values for the class is available via $(relationAllInstance \ Q \ O' \ (?V'_{min} \ ?V'_{max}))$
 - Return geometric-mean of $?V'_{min}$ and $?V'_{max}$ as estimate.

Figure 4.5: The ontology heuristic in BotE-Solver

4.3.2 Mereological Estimation Heuristic Method

The mereology heuristic transforms the object using part-whole relationships into other objects for which estimates might be more readily made. The first notion that is key to implementing mereology is knowing whether the quantity is extensive or intensive. An extensive quantity is a physical property that is dependent on the system size, for example, mass, volume, heat, etc.; while an intensive quantity is one independent of system size, for example, density, temperature, melting point, etc. If Q is an extensive parameter, then, $v = \sum v_i$. If O is homogeneous, i.e., composed of the same kind of objects, then the above sum reduces to a product of the number of parts and the value for each part, $v = n * \sum v'$. In some situations, homogeneity can be an assumption to approximate a more complex calculation involving all the subparts. If Q is an intensive parameter like density, we look for the constituents of O . In this case, we need to know all the constituents and for each of them the fraction that they constitute of the whole, then, $v = \sum w_i * v_i$, where w_i is the fraction of the part i .

Cyc does not define notions of extensive and intensive quantities, so we have to add that knowledge. Additionally, there are many different ways in which part-whole knowledge is represented in Cyc: part-whole relationships are represented by `physicalParts`, `systemComponents`, `groupMembers`, `subEvents`, `constituents`, among others. This requires adding heuristic applicability knowledge for each of these types of part-whole knowledge.

```

(defSuggestion HomogenousGroupExtensiveQuantityStrategy
  (valueOf ((QPQuantityFn ?quantity) ?object) ?ans)
  :documentation "If there is a group of n of something,
    and each one of them has a value q,
    then the value for the all of them is n*q
    if q is extensive dimension"
  :test (and (isa ?quantity ExtensiveMeasurableQuantity)
    (isa ?object Group))
  :subgoals ((groupMemberType ?object ?individual-member)
    (valueOf ((QPQuantityFn numberOfGroupMembers)
      ?object) ?count)
    (valueOf ((QPQuantityFn ?quantity)
      ?individual-member) ?individual-value))
  :result-step (evaluate ?ans
    (TimesFn ?count ?individual-value)))

```

Figure 4.6 The HomogenousGroupExtensiveQuantityStrategy suggestion

Figure 4.6 shows the HomogenousGroupExtensiveQuantityStrategy suggestion, which estimates a quantity for a group by multiplying the value for an individual member by the number of group members. This assumes that the quantity is uniformly distributed, and is a good approximation for estimating things like number of cars or pianos owned by a population. In cases where all members can be enumerated, another variation of this strategy finds the value for each of the members and then adds them up. The enumeration strategy requires making a closed world assumption. In practice, the homogeneous assumption is more widely applicable and frequently employed by human estimators [Swartz, 2003].

```

(defSuggestion CountViaConstituentStrategy
  (valueOf ((QPQuantityFn ?count-quantity)
            ?whole ?part) ?count)
  :documentation "If ?whole is made out of ?part,
                  then the quantities that describes
                  the dimension of the constitution
                  can be used to measure the count"
  :test (and (constituents ?whole ?part)
             (isa ?count-quantity CountQuantitySlot))
  :subgoals ((constituentPhysicalQuantity
              ?whole ?part ?constituent-quantity)
             (valueOf ((QPQuantityFn ?constituent-quantity)
                       ?whole) ?whole-measure)
             (valueOf ((QPQuantityFn ?constituent-quantity)
                       ?part) ?part-measure))
  :result-step (evaluate ?count
                   (QuotientFn ?whole-measure ?part-measure)))

```

Figure 4.7: The CountViaConstituentStrategy suggestion

Figure 4.7 shows another variation of the mereology heuristic. The CountViaConstituentStrategy suggestion computes a count quantity, for example the number of electrons released by a 12 Volt car battery. The predicate constituentPhysicalQuantity relates two objects to a quantity that captures the mereological relationship. In the car battery case, the constituentPhysicalQuantity is the chargeOfObject quantity, which allows us to estimate the number of electrons by dividing the charge contained in the battery by the charge of a single electron.

4.3.3 Analogy Heuristic Method

The analogy heuristic is implemented by KNACK, which was described in the previous chapter. Consider a problem like estimating the season points per game scored by Jason Kidd. KNACK

first retrieves similar examples from memory. The value of the quantity for the closest analogue is called the *analogical anchor*. Using linear regression, it then computes the effect of other causally connected quantities to the quantity to be estimated. This is called *causal adjustment*. The final estimate is generated by combining the causal adjustment with the analogical anchor. Using this heuristic requires having a case library of examples. Therefore, we tested KNACK in the domain of basketball statistics. This domain has favorable properties of having a large number of quantities with causal relationships between them, and allowed us to measure the efficacy of the analogical estimates. However, for the domains covered in the Science Olympics corpus, we do not have cases describing situations similar to the problems.

4.3. 4 Density Heuristic Method

The density heuristic suggests estimating a quantity by finding the associated density and extent quantities associated with it. Figure 11 shows the suggestion that describes this heuristic. In this heuristic, we compute a quantity by finding its associated density and extent quantities. This generalized notion of density includes rates and averages with respect to other parameters as density. We encode these relationships via *densityQuantityFor* and *extentQuantityFor* predicates that state the density and extent relationships for various quantities. Once such knowledge is available the suggestion in figure 4.8 below operationalizes this heuristic.


```

(defSuggestion DensityStrategy
  (valueOf ((QPQuantityFn ?quantity) ?object) ?ans)
  :documentation "implements density heuristic"
  :test (and (densityQuantityFor ?quantity ?density-quantity)
             (extentQuantityFor ?density-quantity ?extent-quantity))
  :subgoals ((valueOf ((QPQuantityFn ?density-quantity) ?object) ?density)
             (valueOf ((QPQuantityFn ?extent-quantity) ?object) ?extent))
  :result-step (evaluate ?ans (TimesFn ?density ?extent)))

```

Figure 4.8: The DensityStrategy suggestion

It states that while estimating any quantity for which density and extent quantities are available, we can find the value by multiplying them.

4.3.5 Domain Laws Heuristic Methods

Domain laws suggests quantity transformations, and include both laws of physics as well as rules of thumb. This is an open-ended set, and we had to add suggestions corresponding to relevant phenomena in the question. Figure 4.9 shows a description of $E=mc^2$. This can be improved by explicitly representing the relationship as an equation, so that it can be used to solve for mass or energy. It suffices for the current examples.

```

(defSuggestion TotalEnergyConversionStrategy
  (valueOf ((QPQuantityFn energyProduced) ?event) ?energy)
  :documentation "E=mc^2"
  :test (isa ?event TotalEnergyConversionProcess)
  :subgoals ((objectActedOn ?event ?obj)
    (valueOf ((QPQuantityFn massOfObject) ?obj) ?mass)
    (valueOf ((QPQuantityFn velocityOfObject) Light) ?c))
  :result-step (evaluate ?energy
    (TimesFn ?mass (ExponentFn ?c 2))))

```

Figure 4.9: The TotalEnergyConversionStrategy suggestion

4.3.6 Scale-Up Heuristic Method

The scale-up heuristic relates two different systems and the quantities and objects between them. We borrow the representation of mapping and correspondences from the analogy ontology [Forbus et al., 2002] to describe scale-up. The analogy ontology provides representations for combining analogical processing via structure mapping engine to a reasoning system. In structure-mapping, a mapping consists of a structurally consistent set of correspondences. A correspondence relates an item in base to a item in the target. Items can be entities, expressions, or functors. The relationship `(correspondsInMapping ?m ?b ?t)` indicates that item `?b` corresponds to `?t` in mapping `?m`. This predicate helps us relate entities and relationships between the scale-model and the situation being modeled.

```

(defSuggestion SeekScaleModelStrategy
  (valueOf ((QPQuantityFn ?quantity) ?object ?situation) ?ans)
  :subgoals ((and (correspondsInMapping ?scale-model ?base-object ?object)
    (isa ?scale-model ScaleModel))
    (valueOf ((QPQuantityFn ?quantity) ?base-object ?situation)
      ?base-ans)
    (valueOf ((QPQuantityFn scalingFactor) ?scale-model)
      ?scaling-factor))
  :result-step (evaluate ?ans (TimesFn ?scaling-factor ?base-ans)))

```

Figure 4.10: The SeekScaleModelStrategy suggestion

The suggestion in Figure 4.10 says that while estimating a quantity, if a scale model is available in which the object in question corresponds to something for which we know the value of the quantity, then we look for the scaling factor and multiply by it to generate an estimate. Computing scaling factor is domain specific and are important scientific achievements. For example, the following represents an important result in animal physiology: generally speaking, the energy requirements of an animal vary with the $3/4^{\text{th}}$ power of the surface area [for details, see Schmidt-Nielsen, 1994].

```
(defSuggestion AllometricScalingLaw
  (valueOf ((QPQuantityFn scalingFactor) ?scale-model)
    ?scaling-factor)
  :test (and (isa ?scale-model ScaleModel)
    (scaleModelFor ?scale-model ?quantity)
    (isa ?quantity EnergyQuantitySlot))
  :subgoals ((correspondsInMapping ?scale-model ?base-object
    ?target-object)
    (and (isa ?base-object BiologicalSpecies)
      (isa ?target-object BiologicalSpecies))
    (valueOf ((QPQuantityFn surfaceAreaOfWholeObject)
      ?base-object) ?base-area)
    (valueOf ((QPQuantityFn surfaceAreaOfWholeObject)
      ?target-object) ?target-area))
  :result-step (evaluate ?scaling-factor
    (ExponentFn
      (QuotientFn ?target-area ?base-area) 0.75)))
```

Figure 4.11: The AllometricScalingLaw suggestion

4.3.7 System Laws Heuristic Method

These are domain laws that apply to a system as a whole. This can simplify problem-solving by avoiding introducing unknowns in the problem solving process by considering the system as a whole as opposed to individual objects. Very common in scientific reasoning, we do not often run into this heuristic in BotE reasoning. Figure 16 shows the application of conservation of

linear momentum, such a system law. This is another domain where Cyc is weak in its representation of domain laws required to infer whether quantities are conserved for a system.

```
(defSuggestion BalanceStrategy
  (valueOf ((QPQuantityFn ?quantity) ?system ?state) ?val)
  :test (and (isa ?quantity ConservedQuantity)
             (isa ?system System)
             (quantityConservedForSystem ?quantity ?system))
  :subgoals ((and (stateOfSystem ?system ?alt-state)
                  (different ?state ?alt-state))
             (valueOf ((QPQuantityFn ?quantity)
                       ?system ?alt-state) ?alt-val))
  :result-step (evaluate ?val (IdentityFn ?alt-val)))
```

Figure 4.12: The BalanceStrategy suggestion

4.4 BotE-Solver at the Science Olympics

We first present a table of all the 35 problems and BotE-Solver's answers to each of them. The predicate calculus representation of each question is also shown. The table is followed by a brief discussion of the performance of BotE-Solver.

Table 4.1: The Science Olympics questions and BotE-Solver's answers

Question and the predicate calculus representation	BotE-Solver's answer
When the island of Krakatoa was destroyed by a volcanic eruption, the sound waves could be detected world wide. How long would it take for such a wave to travel around the earth and come back to Krakatoa? (Seconds) (valueOf ((QPQuantityFn duration) KrakatoaWaveAroundEarthScenario26) ?time)	1.2e+5
How long would it take a paramecium to swim from London to Toronto? (Seconds) (valueOf ((QPQuantityFn duration) ParameciumSwimmingScenario11) ?time)	1.2e+10

<p>How many years would it take for McDonalds to sell a mole of their hamburgers? (Years)</p> <p>(valueOf ((QPQuantityFn duration) McDonaldsMoleHamburgerSale1) ?time)</p>	8.7e+13
<p>How many dollars would each person on this planet possess if there were a mole of dollars to distribute? (Dollars)</p> <p>(valueOf ((QPQuantityFn amountOfMoneyTransferredInEvent) Person MoleGiveaway1) ?money)</p>	1.0e+14
<p>People crowd into London until all available open space within the city limits is covered with standing people. How many people would there be? (Number)</p> <p>(valueOf ((QPQuantityFn countContained) Person CrowdedLondonScenario35) ?count)</p>	1.1e+7
<p>How many bricks are there in London? (Number)</p> <p>(valueOf ((QPQuantityFn countContained) Brick GreaterLondon-EnglandRegion) ?count)</p>	3.2e+10
<p>What is the thickness of a sheet of paper in wavelengths of visible light? (Number)</p> <p>(valueOf ((QPQuantityFn ratioOfToAlongQuantity) SheetOfPaper VisibleLight heightOfObject wavelength) ?ans)</p>	299
<p>To what height could loose leaf paper be stacked if you possessed Avogadro's number of sheets? (Meters)</p> <p>(valueOf ((QPQuantityFn totalValue) heightOfObject SheetOfPaper AvogadroSheets1) ?height)</p>	9.5e+19
<p>How many meters would a ground state electron in a hydrogen atom be from its nucleus if the nucleus of the hydrogen atom was blown up to the size of a baseball? (Meters)</p> <p>(valueOf ((QPQuantityFn distanceBetween) (ScaleModelAnalogueFn HydrogenAtomBaseballModel16 (NucleusFn (AtomFn Hydrogen))) (ScaleModelAnalogueFn HydrogenAtomBaseballModel16 Electron)) ?distance)</p>	3.7e+3
<p>How many electrons could a fully charged 12 volt car battery release before it was completely discharged? (Number)</p> <p>(valueOf ((QPQuantityFn hasPhysicalPartCount) AutomobileBattery Electron) ?count)</p>	4.4e+24
<p>How many electrons are there in the electron beam between the cathode of your T.V. set and the screen? (Number)</p> <p>(valueOf ((QPQuantityFn hasPhysicalPartCount) TelevisionCathodeRayTube1</p>	4.4e+17

CathodeRayTubeElectron) ?count)	
How many piano tuners are there in Toronto? (Number) (valueOf ((QPQuantityFn countContained) PianoTuner CityOfTorontoOntario) ?count)	136
How many automobiles are scrapped in the USA per year? (Number) (valueOf ((QPQuantityFn countDiscardedPerYear) Automobile UnitedStatesOfAmerica) ?count)	6.7e+6
How many photons/sec are emitted by a 100 watt light bulb? (Number) (valueOf ((QPQuantityFn countContained) Photon LightBulbEvent1) ?ans)	2.5e+20
How many kilometers of D.N.A. are there in the cells of one human body? (Kilometers) (valueOf ((QPQuantityFn totalValue) lengthOfObject DNAStrand HumanBody) ?length)	2.3e+14
How many sodium ions are in one tablespoon of salt? (Number) (valueOf ((QPQuantityFn hasPhysicalPartCount) TablespoonOfSalt24 (IonFn Sodium)) ?count)	9.9e+18
How many gas molecules are there in the earth's atmosphere? (Number) (valueOf ((QPQuantityFn hasPhysicalPartCount) TheEarthsAtmosphere (MoleculeFn Air)) ?count)	9.1e+18/1.5e -24 = 6.1e+42 ²⁸
How many atoms of iron are there in a sewing needle? (Number) (valueOf ((QPQuantityFn hasPhysicalPartCount) SewingNeedle (AtomFn Iron)) ?count)	3.5e+19
How many oxygen molecules enter your lungs on each inhalation? (Number) (valueOf ((QPQuantityFn hasPhysicalPartCount) (InhalingFn Oxygen) (MoleculeFn Oxygen)) ?count)	6.7e+16
How many water molecules are there in a totally filled olympic size swimming pool? (Number) (valueOf ((QPQuantityFn hasPhysicalPartCount) SwimmingPool1 (MoleculeFn Water)) ?count)	2.6e+27
How many air molecules in an automobile tire? (Number) (valueOf ((QPQuantityFn hasPhysicalPartCount) AutomobileTire (MoleculeFn Air)) ?count)	1.3e+23
How many cells are there in the human body? (Number) (valueOf ((QPQuantityFn hasPhysicalPartCount) HumanBody Cell) ?count)	9.5e+13
If fighter pilots experience too high “gee” forces in a turn they black out. What is the minimum safe vertical turning circle for a plane travelling at the speed of sound? (Meters)	1.3e+5

²⁸ Allegro Common Lisp is not able to compute this quantity and returns #.excl.:*infinity-single*, this answer is generated from BotE-Solver's subgoals of volume of earths atmosphere and volume of the air molecule that were used to generate the answer.

(valueOf ((QPQuantityFn radiusOfTurn) FighterPilotScenario27) ?radius)	
Calculate the gravitational attraction between a man and a woman as they stand talking to each other. (Newtons) (valueOf ((QPQuantityFn gravitationalForceBetween) Man32 Woman32) ?force)	4.5e-7
How many joules of chemical energy are there in one litre of gasoline? (Joules) (valueOf ((QPQuantityFn chemicalEnergyContent) Gasoline) ?energy)	1.9e+8
What is the mass of earth's population? (Kilograms) (valueOf ((QPQuantityFn massOfObject) HumanPopulationPlanetEarth) ?mass)	5.2e+11
What is the weight of the air over Lake Superior? (Kilograms) (valueOf ((QPQuantityFn massOfObject) AirOverLakeSuperior) ?x)	1.8e+15
How much energy does a horse consume in its lifetime? (Joules) (valueOf ((QPQuantityFn energyConsumptionDuring) Horse1 HorseLifetime1) ?x)	2.5e+7
How much does the Thames River heat up in going over the Fanshawe Dam? (Degree Celsius) (valueOf ((QPQuantityFn changeInQuantity) temperatureOfObject FanshaweDamHeatingEvent) ?increase)	47
How many liters of gasoline are used in Chicago in a year? (Liters) (valueOf ((QPQuantityFn annualConsumption) Gasoline ChicagoDrivingScenario21) ?amount)	5.7e+6
Estimate the mass of lead deposited each year in London due to emissions from automobiles. Each litre of gas contains about 2 grams of lead. (Kilograms) (valueOf ((QPQuantityFn emissionAmountDuring) Automobile Lead LondonScenario28) ?lead-amount)	1.5e+4
There are approximately 1.5e+9 cubic kilometres of ocean. If the water was to evaporate, what mass of minerals would remain behind? (Kilograms) (valueOf ((QPQuantityFn totalValue) massOfObject Mineral TheOceanSea) ?amount)	5.2e+16
What volume of hydrogen gas measured at S.T.P. could be produced by the electrolysis of all the water in Lake Erie? (Liters) (valueOf ((QPQuantityFn totalValue) volumeOfObject Hydrogen LakeErie) ?length)	6.0e+14
How much energy is produced by fully converting a teaspoonful of water to energy? (Joules) (valueOf ((QPQuantityFn energyProduced) TeaspoonIntoEnergyEvent1) ?energy)	9.0e+14
How much water can be brought to boil using this amount of energy?	4.0e+11

(Kilograms) (valueOf ((QPQuantityFn amountOfParticipant) WaterHeatingEvent2 Water2) ?water2)	
--	--

For most of these questions, correct answers are not available, therefore it is not straightforward to make claims about the accuracy of the estimates generated by BotE-Solver. Based on quantitative facts and physics formulas from The Physics Factbook²⁹, Eric Weisstein's World of Physics³⁰, and Wikipedia³¹, calculations done by hand by me show that all the answers are in within an order of magnitude of hand generated answers. However, the key claim here is not about the accuracy of answers. The seven classes of heuristic methods described in Chapter 2 were sufficient for solving all of these problems. It should be noted that this evaluation was done *after* we proposed the seven heuristic methods in [Paritosh and Forbus, 2005]. This demonstration also shows that BotE-Solver's architecture has enough expressive power to operationalize the heuristics.

4.5 Discussion and Conclusions

BotE-Solver provides support to the knowledge-level claim that the seven heuristic methods of ontology, similarity, mereology, density, domain laws, balances and scale-up are comprehensive for BotE reasoning. Combined with the evidence from the corpus analysis of worked out problems in Swartz [2003], this provides compelling support for the knowledge-level claim.

However, operationalizing all of those seven heuristic methods in a fully general manner is difficult. A heuristic method in BotE-Solver is implemented via multiple different suggestions.

²⁹ <http://hypertextbook.com/facts/>

³⁰ <http://scienceworld.wolfram.com/physics/>

³¹ <http://en.wikipedia.org/>

The multiple implementations deal with the *representational variance* in the knowledge base. Representational variance is the fact that different predicates (with possibly different argument structures) are used to describe concepts that a problem solver might want to treat alike. For example, the mereological relationship in the characterization of the mereology heuristic is expressed by 520 different predicates in the current version of Cyc KB, some of which are: `subEvents`, `intangibleParts`, `subCultures`, `subSeries`, `capitalCity`, `chapters`, `citizens`. This leads to having different ways to express the same notion of extracting quantity value for each of the subparts and adding them up. Even a simple concept of area has multiple incarnations: `areaOfObject`, `areaOfRegion`, `landAreaOfRegion`, `surfaceAreaOfTopOfObject`, `MetropolitanAreaFn`, to name a few. We are not suggesting that this is a negative issue for the knowledge base, and that it should use the same predicate for all these concepts: these are important distinctions that should be represented in any large, rich knowledge base. Representational variance is inevitable and desirable in a large knowledge base. For a problem solver, especially one like BotE-Solver, which seeks to make educated guesses, it is important to be able to use knowledge which is similar enough, but is differently represented.

One heuristic that is most generally described is the ontology heuristic, which is because the ontology is represented by a small set of predicates like `isa`, `genls`, `relationAllInstance`, among others. An important thing about the ontology predicates is that they apply generally to a large class of objects, while the mereology predicates are more specialized. This is the source of having to write multiple suggestions to represent the same heuristic method. One very promising approach to automatically tackle representational variance

in a problem solver is the idea of learning *domain mappings* by analogy [Klenk and Forbus, 2007].

Chapter 5: Related Work

This research draws upon work in three different areas: heuristics, automated problem solving, and psychology of human reasoning. We discuss each in turn.

5.1 Heuristics

George Polya (1945) popularized heuristics in his book as possible steps one could take while solving mathematical problems. Some of his heuristics included drawing a figure, working backward from what is to be proved and considering a more general version of the problem. The final output in such reasoning is sound mathematical statements, the heuristics help explore the space in a clever way. Herb Simon coined the notion of *Bounded Rationality* and *Satisficing* (1957). In this approach, reasoning is still governed by laws of rationality and realistic resource constraints are placed on it. Newell and Simon (1963) proposed weak methods, e.g., means-end analysis, generate and test, etc., as the basis of intelligence.

Doug Lenat's AM and Eurisko (1982) systems simulated scientific discoveries in the domain of mathematics, device physics, games, and heuristics itself, among others, armed with a library of

hundreds of heuristics. Lenat called for a formal study of the science of heuristics, *heuretics*. However, Lenat's notion of heuristics is different from ours. The goal of his systems was to make interesting scientific conjectures, and his heuristics guided exploring the space. For example, one of his heuristics would suggest that if a function $f(x,y)$ takes two arguments, then it's worth the time and effort to define and explore the behavior of $g(x)=f(x,x)$, that is, to see what happens when the arguments coincide. If f is multiplication, this new function g is squaring; if f is union or intersection, then g is the identity function, and so on. His notion of heuristics was ways to branch out and explore the space in some guided ways. This is different from the way we are framing heuristic reasoning: our goal is to generate answers quickly, rather than explore a space of hypotheses.

Gregg Collins identified a set of two dozen strategies and described how each would be applied in the domain of football [Collins, 1987]. Many of these strategies could be referenced with common phrases, e.g., “hedge your bets”, “hold the fort”, etc. He showed that these were compositional, and argued that these could be applied to other domains. Starting from Collins' work, Andrew Gordon has catalogued commonsense planning strategies in ten diverse domains including business practices, education, Machiavellian politics, personal relationships, among others [Gordon, 2004]. He has identified 372 strategies, and nearly 1000 fundamental concepts, and organized them into different knowledge areas. A caveat is that these strategies are represented in a *pre-formal* representation: they are written in English, where words and phrases that will be needed by any formal reasoning system are marked. For example,

Warfare strategy 44. Give false information to enemy spies: Secretly use enemy spies to send deceptive information to your enemy.

Representation: The planner has an *Adversarial relationship* with another agent A1. An agent A2 has a *Cooperative relationship* with the agent A1 to execute *Cooperative plans* that include *Informing* the agent A1 of *Information* that involves the planner. The agent A2 has the *False belief* that the planner has a *False belief* that the planner has a *Cooperative relationship* with the agent A2. The planner *Monitors* planning for *Adversarial plans* that have a *Threat* that the adversary agent A1 will *Successfully execute* a *Counterplan*. In this case, the planner executes a plan to cause the agent A2 to have a *False belief* about the *Adversarial plan*, and then *Enables* the *Cooperative plan* of this agent A2 that includes *Informing*.

There is significant representational and computational work that needs to be done to get a computer problem solver to use these strategies. However, this approach is compatible with ours and indirectly supports a crucial working assumption: the number of strategies (or heuristics) is not too many. In contrast with the number of facts or rules, which are in millions, these are in hundreds.

5.2 Automated Problem Solving

Problem solving has a very broad meaning in AI. We are interested in problem solving in knowledge-rich scenarios. Below we describe three systems that solve problems in different domains exhibiting interesting levels of competence, comparable to a human.

5.2.1 SAINT

SAINT (Symbolic Auto INTegrator) [Slagle, 1963] solved symbolic integration problems in freshman calculus. It was able to solve 52 out of 54 problems from the MIT freshman calculus final examinations. It solved problems by applying two types of transformations: 1) heuristic transformations, e.g., substitutions, which might or might not succeed; and 2) algorithmic

transformation, e.g., taking the constant out of the integrand. It maintained an AND/OR tree for keeping track of the decomposition introduced by these transformations. It also maintained its agenda in two lists: a temporary goal list, and a heuristic goal list. The design of the AND/OR solver in BotE-Solver is influenced by SAINT and its reconstruction as JSAIN [Forbus and de Kleer, 1993].

5.2.2 FERMI

FERMI (Flexible Expert Reasoner with Multi-domain Inferencing) [Larkin *et al*, 1988] used two general principles, *decomposition* and *invariance*, with domain specific knowledge to solve textbook problems in fluid statics, DC-circuits and centroid location. In this system, the factual and strategic knowledge required for problem solving were stored in two different semantic hierarchies. These hierarchies are isa-hierarchies consisting of schemas. The *decomposition principle* says that

$$Q(E) = \sum Q(E_i)$$

Where $\{E_i\}$ are the individual components of E, and Q is a quantity that is decomposable with respect to the entity E. For example, the area of a surface can be computed by summing up areas of all the components. This is a special case of the mereology strategy in BotE-Solver. The *comparison of invariants* principle says that if a quantity Q is invariant with respect on an entity. This is stated in FERMI as

$$Q(E_i) = \text{constant for all } E_i$$

The subscript in this equation refers to different states of E (as opposed to the decomposition principle where it refers to sub-components of E). This results in an equation like $Q(E_1) = Q(E_2)$

where E_1 and E_2 are two different states of E . FERMI's reasoning ability was the application of these two principles. These principles were implemented specifically in three domains: 1) computing pressure drops in fluids at rest, 2) potential drops in DC-circuits, and 3) calculation of centers of mass for planar objects.

FERMI is unfortunately named, as it doesn't and wouldn't be able to solve open ended problems like the Fermi problems, given its reasoning mechanisms. BotE-Solver implements the two principles in FERMI, and five more that are more useful in solving Fermi problems.

5.2.3 TPS

TPS (Thermodynamics Problem Solver) [Pisan, 1998] solved over 150 thermodynamics problems taken from multiple thermodynamics textbooks and produced expert-like solutions. The underlying architecture of TPS, IPSA (Integrated Problem Solving Architecture) combined qualitative, quantitative, and diagrammatic reasoning. TPS used qualitative representations to represent modeling assumptions and causal knowledge required for problem solving. It used an extended version of the planning framework of the RAPS system [Firby, 1989] to represent problem solving methods. There were two types of plans in TPS: goal-centered and problem-centered. The plans provided knowledge for typical ways of solving problems, grouping similar methods for achieving the same goal, and representing connections between domain primitives. It used suggestions, similar to SAINT and BotE-Solver, where plans are a type of suggestions that are given priority over others like equation solving and table-lookup. An important contribution of TPS was analysis of functional classes of equations in reasoning about thermodynamics problems.

5.2.4 Semi-quantitative Reasoning

There has been important work in the QR community in combining qualitative reasoning with quantitative knowledge. It is important to distinguish between the notion of quantitateness in semi-quantitative reasoning [Berleant and Kuipers, 1997] and BotE reasoning. In semi-quantitative reasoning, functional uncertainty is represented by defining envelopes within which functional constraints must lie, and parametric uncertainty is represented by numeric intervals. Clearly, this is still in the spirit of purely first-principles reasoning, in contrast to our heuristic approach.

5.3 Psychology of Human Reasoning

The flexibility and robustness of human reasoning is an important motivation for our work. We believe that human reasoning provides constraints for organization of knowledge and heuristics that are useful for building robust problem solvers.

5.3.1 Plausible Reasoning

By analyzing verbal protocols of people's answers to everyday questions, Allan Collins constructed a theory of human plausible reasoning [Collins, 1978a, 1978b]. One part of this theory was a set of plausible reasoning inference patterns. A key difference from logical reasoning was that the theory specified how different information in memory affects the certainty of the conclusions drawn. Here are some examples of the type of protocols that went into this theory:

Q: Is Uruguay in the Andes Mountains?

R: I get mixed up on a lot of South American countries (pause). I'm not even sure. I forget where Uruguay is in South America. It's a good guess to say that it's in the Andes Mountains because a lot of the countries are.

Q: Do you think they might grow rice in Florida?

R: Yeah, I guess they could, if there were an adequate fresh water supply. Certainly, a nice, big, warm, flat area.

The four major types of plausible inference patterns were: *Generalization*, *Specialization*, *Similarity* and *Dissimilarity*. BotE-Solver implements the first three of these. Reasoning via dissimilarity allows inferring that an object might not have a property since it is dissimilar to others that have that property. This is useful in classifying things. For example, one might conclude that coffee is not grown in Russia based on its dissimilarity to other coffee growing countries (they are much warmer, e.g., Ethiopia, Brazil, India). Surprisingly, dissimilarity is rarely used in BotE reasoning. Collins' work is foundational, and as we explore other domains of heuristic reasoning besides BotE, it will provide a gold mine of constraints for building problem-solving systems.

5.3.2 Heuristics and Biases

In the 1970s, the psychologists Amos Tversky and Daniel Kahneman started the *Heuristics and Biases* program. The goal in this program was to use peoples' systematic biases in judgment under uncertainty to reveal the heuristics they use. Tversky and Kahneman (1974) reported people's assessment of probabilities of uncertain events. In a very important set of results, they show that people make systematic errors because of a set of heuristics that they employ.

An important psychological heuristic is the *availability heuristic* (Tversky and Kahneman, 1973). According to the availability heuristic, the ease with which instances come to mind is used as indicator of the size or frequency of the class. For example, when asked the question, “Do homicides or suicides cause more deaths in the US?” most people erroneously answer homicides, as it is easier to recall examples of homicides than suicides. Tversky and Kahneman's goal was to highlight the heuristic by pointing out when it leads to systematic errors. However, the availability heuristic is usually reasonable. Another heuristic is the *representativeness* heuristic says that people judge the probability that P is a member of category C on the basis of the similarity of P to our concept of a prototypical member of C. This work has led to a large body of literature in Psychology exploring various aspects of intuitive reasoning in judgment and decision making.

5.3.3 Simple Heuristics that Make Us Smart

Gerd Gigerenzer and his group (1999) have made compelling arguments for *fast and frugal heuristics*, in which they view the mind having an adaptive toolbox of heuristics that work because of the way the environment is structured. One of their heuristics is the *recognition* heuristic: something that you can recognize is likely more important than something you don't. In a study where both a sample of German and US students were asked questions about cities like “Which is bigger: San Antonio or San Diego?” they showed that Germans performed significantly better than Americans on American cities and vice versa for German cities. Their argument is that with lesser knowledge of American cities, German students can invoke the

recognition heuristic to pick the answer that is most likely going to be right, while American students cannot use that heuristic as they probably have heard of both cities. However, their focus is on populating this toolbox and not on figuring out how this might be integrated with other cognitive functions.

Other heuristics proposed by Gigerenzer are: *take-the-first* and *take-the-best* heuristics, which suggest that even though we need to know information along various dimensions to predict if a country is a developing nation, usually we can make a decision based on just one dimension. They argue that this is owing to the non-compensatory nature of cues in the world, which says that the classification made using the most important dimension is likely to be right, as that dimension usually dominates all the other dimensions.

5.3.4 Education

Linder (1999) studied quantitative estimation in the context of engineering education. About a hundred mechanical engineering seniors at MIT, and fifty each at five other universities attempted these estimation questions. He also compiled responses from a hundred professionals, out of which about there were about thirty each of electrical and mechanical engineers, and the rest from other engineering and science backgrounds. Based on these verbal protocols he tried to build a framework for how people do rough estimations. His focus was how to improve engineering curricula, and thus his framework is informal and not couched in computational terms; nevertheless, it provides an interesting source of data. In one experiment, when people were asked to estimate dimensions of an aluminum bar, more than 50% came up with correct estimates and all the answers were in the correct order of magnitude. However, in the same

experiment, more than 90% of mechanical engineering seniors (100 at MIT, and 250 from five other universities) came up with wrong order of magnitude estimates of value of energy stored in a 9-volt “transistor” battery [Linder, 1999]. The responses varied by nine orders of magnitude excluding outliers! The heuristics presented in this thesis could be used to develop a curriculum of estimation instruction.

Chapter 6: Conclusions and Future Work

Human reasoning is robust and flexible, while most AI systems are fragile and brittle. The sources of brittleness in most AI systems appear to be gaps and inaccuracies in knowledge, and inferential complexity. On the other hand, flexibility in human reasoning arises in part from the ability to come up with plausible answers, educated guesses, and reasonable explanations. This is the intuition behind the heuristic reasoning approach proposed in this thesis. This approach suggests *reasonableness*, in quality of answer, and *comprehensiveness*, broad coverage in the task domain, instead of soundness and completeness. BotE-Solver demonstrates this approach in the domain of *Back of the Envelope* (BotE) reasoning. Armed with a rich ontology, analogy and a library of heuristic methods, the system can generate reasonable answers to a broad set of questions.

Chapter 2 described the BotE reasoning domain, a formal representation for the problems and heuristic methods, and a corpus study of BotE problems. Chapter 3 described a theory of representation and learning about quantities and the CARVE system. Chapter 4 described the BotE-Solver system, illustrated its operators via examples, and evaluated its performance on the Science Olympics corpus. Chapter 5 described related work. The next section summarizes the major contributions of this thesis. We then describe some of the promising future directions for this work.

6.1 Summary of Key Contributions

6.1.1 A Broad Coverage Theory of Back of the Envelope (BotE) Reasoning

The goal in BotE reasoning is to produce a rough quantitative estimate. Given a question, the first step is to see if one directly knows the answer or knows similar examples for which estimates are already available. This step is called *direct estimation*. Indeed, if one had no quantitative knowledge at all, it will be impossible to answer such questions. However, it is not reasonable to assume direct access to all quantitative facts. What makes BotE reasoning powerful is the fact that by piecing together simple facts that are readily available, one can answer seemingly difficult questions. This step is called *estimation modeling* [Paritosh and Forbus, 2003]. Estimation modeling is the process of constructing simplified models of complex scenarios which are good enough for the purposes of making a rough estimate. These models are constructed by applying heuristic methods to transform the original problem into other problems which are possibly easier.

We present a set of seven heuristic methods for estimation modeling: analogy, ontology, mereology, density, domain laws, system laws, balances and scale-up [Paritosh and Forbus, 2005]. These heuristic methods are implemented in BotE-Solver. We show two-fold support for the comprehensiveness of this library of heuristic methods. In a corpus analysis of problems from Swartz's "Back-of-the-Envelope Physics," we found that the above seven heuristic methods accounted for 94% of the strategy use. The remaining 6% contain instances of designing experiments to estimate a quantity, and one instance of a complex problem from statistical

mechanics. This corpus analysis is described in Chapter 2. Further, these methods are sufficient for BotE-Solver to solve all the thirty five Fermi Problems from the Science Olympics.

6.1.2 A Cognitively Plausible Theory of Learning about Quantities

To get to a quantitative estimate the reasoning process has to bottom out by plugging in the numeric values. Humans get better at solving BotE problems by exposure to more quantitative facts in the domain [Linder, 1999]. We call this facility with quantitative knowledge built out of experience as *quantity sense*. We present a theory of quantity sense that answers two questions: 1) What do people learn about quantities?, and, 2) How do people learn about quantities?

We introduce the *symbolization by comparison (SBC)* theory [Paritosh, 2004; forthcoming]. The SBC theory claims that people's knowledge about quantities consists of a symbolization of the continuous quantity which is built by processes of comparison. Comparison helps us notice and extract the scale of values of quantities and we create symbolizations that name points and intervals on this scale. A *symbolization* is a qualitative representation of a continuous quantity. These symbolizations must make two kinds of distinctions: *distributional*, those that denote changes of quantity, e.g., large and small; and *structural*, those that denote changes of quality, e.g., boiling point and poverty line. Chapter 3 presented evidence from psychology and linguistics, and arguments from ecological and task/reasoning constraints that support the SBC theory. We described CARVE [Paritosh, 2003], a computational instantiation of the SBC hypothesis. CARVE learns qualitative representations of quantity from exposure to examples.

6.1.3 A Theory of Analogical Estimation

Analogical estimation involves using a similar example to make a numeric estimate. For example, the price of a used car might be similar to another car of the same make and mileage. In order to use analogies to make numeric estimates, our analogical matching algorithms must be sensitive to quantities. Most models of similarity do not adequately handle numeric properties – either ad hoc similarity metrics such as Euclidean distance are used, or the quantities are completely ignored in the matching and retrieval processes. The SBC theory presents a different approach to the problem of incorporating quantities in similarity models by proposing that the solution lies in better representations. The qualitative representations generated by CARVE are added to the descriptions being compared, which allows Structure Mapping Theory to be sensitive to quantities. The analogical estimation task gives us a way to functionally evaluate the representations generated by CARVE. In Chapter 3, we showed that representations generated by CARVE lead to more accurate estimates in an analogical estimation task.

Much of psychological research on estimation follows the anchoring and adjustment paradigm [Tversky and Kahneman, 1974] in knowledge-impooverished domains. To observe how experts utilize similarity and causal relationships in real world estimation tasks, we collected verbal protocols of experts doing realistic estimation tasks [Paritosh and Klenk, 2006]. For example, when trying to estimate the rent for an apartment, one might retrieve from memory a similar apartment in the same neighborhood. The value from the analogical reminding serves as an *analogical anchor*. As a first pass, this analogical anchor is evaluated for its plausibility for the value sought. Analysis of the comparison between the problem and the reminding provides the grist for computing *causal adjustments* from the anchor to improve the estimate: for example,

one might notice that the apartment that they were reminded of is smaller, and is in a slightly less desirable location. KNACK is a computational model of this theory of analogical adjustments. The representations generated by CARVE led to more accurate estimates by KNACK.

This account makes some novel psychologically testable predictions: 1) Causal adjustments need not be insufficient and should be correlated with the perceived strength of the causal relationship, 2) Quantitative difference should be judged in proportion to the depth of nested relationships involving the quantity. The verbal protocols are just a start; more psychological evidence needs to be gathered to explore these claims.

6.1.4 Summary

We presented a computational theory and model of BotE reasoning that can successfully solve problems from the Science Olympics. We also presented additional support of the generality and coverage of heuristic methods by a corpus analysis of problems from Clifford Swartz's Back-of-the-Envelope Physics. We believe that this supports the heuristic reasoning approach to alleviating brittleness in AI systems.

An important heuristic method was the use of analogy. In order for the analogical mechanism to capture the role of quantitative knowledge in computing similarity, we implemented a cognitively plausible learning mechanism that automatically builds qualitative representations of continuous quantity by exposure to examples. The model of analogical estimation is compatible with verbal protocols of expert estimators.

6.2 Future Work

6.2.1 Applications

Some immediate applications of BotE reasoning are to build tools for everyday numeracy support, for example, *number-checker-and-explainer* not unlike the spell-checker that can alert and provide explanations when a number does not make sense, or a search engine that is geared towards finding and generating numerical estimates.

Imagine being able to click on a number in a news article or a financial report, and being offered a back-of-the-envelope estimate showing how it makes sense, or being shown other comparable quantities that contextualize it. A similar project was PLUM [Elo, 1996] which augmented news on world-wide natural disasters that readers often find remote and irrelevant. Using community profiles, PLUM automatically compares facts reported in an article to the reader's home community. The reader, browsing through annotations which PLUM generates, discovers, for example, the number of people affected by the disaster as a percentage of the home town population. The BotE heuristic methods offer more general relationships between quantities and can provide much more powerful explanations. These techniques could be used to build both end-user tools and middleware for information extraction and knowledge acquisition, where sanity-checking is an important issue.

6.2.2 Natural Language and Question Answering

Applications like the everyday numeracy tools mentioned above need access to vast amounts of quantitative information across multiple domains. This requires building support for

dynamically extracting representations from natural language sources like books, newspapers and the web. The extraction and learning of verifiable quantitative facts is an open research question, but one that seems tractable. First, a majority of quantitative information has a small set of linguistic expressions [Schwarz, 1996; Kuehne, 2004]. Second, if one has access to highly redundant corpora like web, numeric information is easier to compare and cluster, which provides one mechanism for verification.

Answering questions is a common task for many different paradigms of AI research. The knowledge-based approach focuses on *problem solving*, and is accomplished by reasoning with formal representations provided with the problem and/or available in a knowledge base. The text-based approach focuses on *information extraction*, and is accomplished by retrieving and analyzing relevant text documents from a corpus.

Both of these approaches have strengths and weaknesses. The knowledge-based approach uses formal representations which allow sophisticated inference and the ability to provide proofs as explanations for solutions. However, the cost of knowledge representation is steep: in a recent evaluation³², it was estimated that it costs about \$10,000 to encode one page of high school chemistry textbook. The amount of knowledge required to successfully answer a broad range of questions like those in the TREC question answering track is vast. To our knowledge, no fully knowledge-based systems have been fielded in the TREC competitions. The text-based approach short-circuits the knowledge issue, and can directly tap into a vast text corpora: web pages, newspaper articles and scientific papers, to name a few. However, the text-based approach has very little if any capability to produce explanations, sanity-check answers and make inferences. Consider the answer of 360 tons for the question “How much Folic acid should an expectant

³² <http://www.projecthalo.com/>

mother consume per day?” Simple chains of reasoning can reject this answer, but most current text-based approaches cannot do this.

The strength of the KB approach is the weakness of the TB approach and vice versa. We believe a key piece of the puzzle in integrating these approaches is heuristic reasoning. Starting with a domain like BotE reasoning, where the types of heuristic methods are well understood, we can build the next generation of question answering systems that leverage the best of both KB and TB approaches. We describe a preliminary analysis of such integration and linguistic expressions for BotE questions and heuristics in Paritosh [2007].

6.2.3 Heuristic Reasoning

BotE reasoning is an instance of heuristic reasoning. This raises questions like: What are other domains of heuristic reasoning? What are the general aspects and properties of heuristic reasoning?

Heuristic reasoning methods exploit the information processing structure of the reasoning system and the structure of the environment to produce reasonable answers when knowledge and/or computational resources for finding the perfect correct answer might not exist. Capturing all the heuristics to generate reasonable answers might not be as colossal of a project as it might first seem: we conjecture that there are about fifteen heuristic domains, and each of them have approximately ten heuristic methods [Paritosh, 2006]. The figure of fifteen is not sacrosanct, it is based on our efforts to build an exhaustive list from analysis of problem solving in multiple domains and the literature on psychology of human problem solving, judgment and decision making. Let's consider a non-BotE example of a heuristic method. Suppose you were asked,

“What American company sells the most greeting cards?” One way to answer the question might be to look up statistics about sales of various greeting card companies. However, a typical human answer might look more like the following:

“Let's see... Hallmark comes to mind. I have seen Hallmark cards all over the place. In fact, I can't think of any other major greeting card manufacturer, so I bet it's Hallmark.”

The above answer and rationale appear reasonable to most people, and in most circumstances such reasoning is right³³. It exploits an important fact about human memory: the ease with which we can recall instances of something is usually correlated with the frequency of that thing in the world, and unheard-of things are often not very important. Reasoning tasks where there are multiple answers and/or processes to arrive at the answer, with varying degrees of correctness or quality are heuristic domains. On the other hand, questions like “What two US biochemists won the Nobel prize in 1992?” or “What is the scientific name of Viagra?” are examples for which it is less likely to have reasonable guesses – you either know the answer or don't. Both of these questions are from the TREC³⁴ corpus, which places more emphasis on such questions than on those that require reasoning/inference. Next we present a set of eight other heuristic domains.

6.2.3.1 Heuristic Domains

In this section we present a list of heuristic domains, and some hypotheses about heuristic methods that might work in those domains.

³³ Hallmark's revenue is approximately \$5 billion, its rival American Greetings' revenue is around \$2 billion.

³⁴ <http://trec.nist.gov/>

Temporal Estimation: When did X happen?

Even when we do not know the exact date when something happened, research in autobiographical memory (Thompson et al., 1996) suggests that by recalling landmark events and constructing a local temporal scale, people can generate reasonable estimates. Allen's temporal interval calculus (1983) presents a concise set of relationships that could be used to organize the heuristic methods in this domain. For example, consider various ways to answer “When was Mark Twain born?” If you happened to know that Mark Twain wrote a first-person account of his participation³⁵ in the American Civil War, which went on from 1861 to 1865, then you might guess that he was probably born around 1830.

Comparison: Is X larger than Y along dimension D? Who/What is the maximum/minimum of a class/set along dimension D?

These questions involve making comparisons between two or more objects along some scalar dimension. At first glance, this might look like solving a few back of the envelope problems and comparing the results. However, it is often easier to answer the comparative question. For example, it is easier to say that Microsoft research spending is more than Apple's than it is to estimate their respective spendings and compare them. One heuristic method here is projection. If we are comparing X and Y along dimension D, and we know another dimension E that is qualitatively proportional to D, then we can project the ordinal result along E on to D. Qualitative representations and techniques of comparative analysis (Weld 1987) might play an important role in this heuristic domain. An important psychological heuristic method is the availability heuristic (Tversky and Kahneman, 1973). According to the availability heuristic, the

³⁵ “The Private History of a Campaign That Failed” also made into a movie.

ease with which instances come to mind is used as indicator of the size or frequency of the class. For example, when asked the question, “Do homicides or suicides cause more deaths in the US?” most people erroneously answer homicides, as it is easier to recall examples of homicides than suicides. Tversky and Kahneman's goal was to highlight the heuristic by pointing out when it leads to systematic errors. However, the availability heuristic is a useful one, and how often it is right is an empirical question. An interesting implication is the idea of “ease of recall”– for most knowledge based systems, fact lookup will take roughly the same amount of time, irrespective of the fact in question³⁶.

Probability: How likely is X? Is X more likely than Y?

People make judgments and decisions based on the likelihood of various events, for example, the author of a scientific paper might consider: “What is the likelihood that my paper will get accepted by a certain conference or journal?” One can generate a reasonable guess about which of two journals are more likely to accept the paper without knowing detailed joint probability distributions. It might be possible to answer the question without knowing a priori all the relevant variables affecting acceptance. One psychological heuristic method to answer these questions is the *representativeness* heuristic (Tversky and Kahneman, 1972) that guides people's estimates of such likelihood. The representativeness heuristic says that people judge the probability that P is a member of category C on the basis of the similarity of P to our concept of a prototypical member of C. Models of analogy and generalization (Falkenhainer, Forbus and Gentner, 1989; Kuehne et al., 2000) could be used to model the representativeness heuristic.

³⁶ With the exception of ACT-R, which takes reaction times as an essential element of modeling. However, ACT-R does not answer the specific questions raised here.

Recent work by Halstead (2005) has incorporated probability into the structured models of generalization.

Classification: Does X belong to the class Y? Does X satisfy property P?

Allan Collins' seminal work on plausible reasoning (1989) gives us a set of strategies used by people in answering such questions, based on an analysis of verbal protocols used by people in answering such questions. Consider questions like: Is Somalia a developing nation? Do they grow coffee in Russia? One could use Somalia's similarity to other instances of developing nation as evidence for answering the question in the affirmative. By noticing the dissimilarities between Russia and other coffee growing countries like Ethiopia, Brazil, Kenya, India, etc., one might conclude that Russia doesn't grow coffee. The representativeness heuristic is useful in answering classification questions as well. Gigerenzer (1999) has proposed the take-the-first and take-the-best heuristics, which suggest that even though we need to know information along various dimensions to predict if a country is a developing nation, usually we can make a decision based on just one dimension. This is owing to the non-compensatory nature of cues in the world, which says that the classification made using the most important dimension is likely to be right, as that dimension usually dominates all the other dimensions.

Choice, evaluation, decision making: Is X good? Is X better than Y? What is the best course of action?

At first blush, this might look like the comparison domain above. However, a key idea in choice and decision making is that of evaluating a situation for how good it is. In Economics, this idea

of evaluation is captured by utility. Prospect theory (Kahneman and Tversky, 1979) is the psychological version of the utility theory. Based on studying firefighters, pilots, nurses in Neonatal Intensive Care Units, and other people who constantly are making decisions with important consequences, Gary Klein (1999) has developed the Recognition-primed decision model, which is essentially an analogical approach. Consider questions like: Is Toyota Corolla the right car for me? Should we hire X or Y? Similarity and experiential knowledge are key elements of the heuristic methods in this domain.

Prediction: What will happen if X?

Qualitative representations and methods of qualitative reasoning are a crucial part of making predictions in the face of incomplete knowledge. Consider: What will happen if the price of gasoline increases? What will happen to the outside temperature if it is snowing? The former involves identifying the causally related quantities to the price of gasoline, and might be explained to a large extent by first-principles qualitative reasoning. However, a more reasonable account of how people might answer the latter question is with experience: we know that it gets relatively warmer after snowing, but might not have a full causal account of the phenomenon. This is the hybrid explanation of qualitative mental models: relying on mostly similarity-based reasoning and only a little on first-principles based reasoning (Forbus and Genter, 1997). It is currently being explored by Yan and Forbus (2004).

Explanation: Why X?

As qualitative representations make causal relationships and modeling assumptions explicit, they naturally provide the grist for generating explanations (Bouwer and Bredeweg, 1999). Consider a question like: Why are hybrid cars more fuel efficient?

Sanity checking: Does X make sense? Is X reasonable?

This is a meta-heuristic domain, where rather than answering a question, we are given a question and a candidate answer, and we use all the above methods to figure out if the answer sounds reasonable. It might be possible to do sanity checking for reasoning domains for which we don't even have heuristic methods. For example, the question in the introduction that asked for the scientific name of Viagra: we can easily reject “Cialis,” “sex,” or “42” as being obviously incorrect. The first step in sanity checking is typechecking – making sure that the candidate answer is of expected class. Maintaining some global sense of various scales is another important aspect of sanity checking. For example, it is easy to reject 14ft as the diameter of Earth. All of the heuristic methods above can be then used to generate a plausible answer and compare it with the candidate answer to conclude if something makes sense or not.

6.2.3.2 Summary

While an ambitious proposal, the decomposition of problem solving tasks into heuristic domains suggests a tractable approach towards building a comprehensive theory and implementation of heuristic reasoning. There are many interesting questions about the nature of heuristic domains that this research program hopes to answer. Which domains and tasks are inherently brittle, and

which domains are heuristic? Are there different types of heuristic domains? We believe that this approach to heuristic reasoning will lead to software that is less brittle, and help us understand the aspects of intuitive reasoning in human minds.

6.2.4 Cognitive and Educational Implications

The *symbolization by comparison* theory has ramifications for education and cognitive theories of numeracy. Recent work has shown that humans along with many other animals share a cognitive infrastructure for representation of approximate numerosity. However, there exists an explanatory gap between how our qualitative representations of quantity are related to this cognitive infrastructure. Quantitative literacy is a very important issue for math education, and there are many demonstrations of the lack of success of current educational methods to impart this skill at various levels from middle-school to college. For example, more than 90% of mechanical engineering seniors (100 at MIT, and 250 from five other universities) came up with estimates that were off by more than one order of magnitude for the value of energy stored in a 9-volt “transistor” battery, and responses varied by nine orders of magnitude. The cognitive insights from this work can be used to design an undergraduate-level class centered on teaching estimation skills. The back of the envelope reasoning provides a framework to structure the class, and this class will also serve as a laboratory to generate and explore hypotheses about human commonsense reasoning.

6.3 Final Words

We have shown in this thesis that a small set of heuristics provide broad coverage in back of the envelope reasoning. BotE-Solver demonstrates the feasibility and the potential of the heuristic reasoning approach. The symbolization by comparison theory, implemented as CARVE, offers a cognitively plausible account of quantity sense.

By leveraging natural language, we can enhance the breadth of BotE reasoning and make tools for everyday numeracy support. The heuristic reasoning research program appears to provide a tractable approach to addressing brittleness in AI systems. Modeling the ability to make educated guesses, which we think is at the heart of human intelligence, could solve the garbage in/garbage out problem.

References

- Ahn, W., Kim, N. S., Lassaline, M. E., & Dennis, M., 2000. Causal status as a determinant of feature centrality. *Cognitive Psychology*, 41, 361-416.
- Allen, James F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11) pp.832-843.
- Ariely, D. (2001). Seeing Sets: Representation by statistical properties. *Psychological Science*, 12(2), 157-162.
- Ashley, K.D. (1990). *Modeling Legal Argument*, MIT Press, MA.
- Banks W. P., and Flora J. (1977). Semantic and Perceptual Processes in Symbolic Comparisons. *Journal of Experimental Psychology: Human Perception and Performance*, 3, 278-290.
- Barbier, V. Grau, B., Ligozat, A., Robba, A. and Vilnat, A., 2005, Semantic Knowledge in Question Answering Systems, *Proceedings of the Knowledge and Reasoning for Answering Questions workshop*, Edinburgh. pp.49-56.
- Berleant, D., and Kuipers, B. 1997. Qualitative and quantitative simulation: bridging the gap. *Artificial Intelligence Journal*, 95(2): 215-255.
- Bierwisch, M. (1967). Some Semantic Universals of German Adjectivals. *Foundations of Language*, 3, 1-36.
- Bobrow, D. G. (1968). Natural language input for a computer problem-solving system, in *Semantic information processing*. Cambridge, Mass.: MIT Press, 146-226.
- Bouwer, A. and Bredeweg, B. 1999. Explanation and Qualitative Reasoning, In *Proceedings of International Workshop on Qualitative Reasoning*.
- Brown, D. R. (1953). Stimulus-similarity and anchoring of subjective scales, *American Journal of Psychology*, 66, 199-214.
- Brown, N.R. (1990). Organization of public events in long-term memory. *Journal of Experimental Psychology: General*, 119, 297-314.
- Brown, N. R., & Siegler, R. S. (1993). Metrics and mappings: A framework for understanding real-world quantitative estimation. *Psychological Review*, 100(3), 511-534.

- Brown, N.R and Siegler, R.S. (2001). Seeds aren't anchors. *Memory and Cognition*, 29(3), 405-412.
- Cech, C. G. and Shoben, E. J. (1985). Context Effects in Symbolic Magnitude Comparisons. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 11, 299-315.
- Chapman, G.B. and Johnson, E.J. (1999), Anchoring, activation and the construction of values. *Organizational Behavior and Human Decision Processes*, 79(2), 115-153.
- Collins, A. and Michalski, R. 1989. *The Logic of Plausible Reasoning: A Core Theory*. *Cognitive Science*, 13, 1-49.
- Cockroft, W.H.: 1982, *Mathematics Counts*, Department of Education and Science (Committee of Inquiry into the Teaching of Mathematics in Schools), Her Majesty's Stationery Office, London.
- Dagan, I., Glickman, O., and Magnini, B., 2006, *The PASCAL Recognising Textual Entailment Challenge*. *Lecture Notes in Computer Science*, Volume 3944, pp 177 - 190.
- Davidson, D. 1967. *The logical form of action sentences* in N.Rescher (ed.). *The Logic of Decision and Action*, U.Pittsburgh Press, 1967.
- Dehaene, S. (1999). *The number sense: how the mind creates mathematics*. New York: Penguin.
- Elo, S. K. (1996). *PLUM: Contextualizing News For Communities Through Augmentation*, Masters Thesis, Program in Media Arts and Sciences, MIT.
- Ericsson, K. A., Simon, H. A. (1993). *Protocol Analysis: Verbal Reports as Data*. MIT Press, Cambridge, MA
- Falkenhainer, B. and Forbus, K. "Compositional Modeling: Finding the Right Model for the Job", *Artificial Intelligence*, 51 (1-3), October, 1991.
- Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41, 1-63.
- Forbus, K. D. (1984). Qualitative process theory. *Journal of Artificial Intelligence*, 24, 85-168.
- Forbus, K. and Gentner, D. (1997). Qualitative mental models: Simulations or memories? *Proceedings of the Eleventh International Workshop on Qualitative Reasoning*, Cortona, Italy.
- Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19(2), 141-205.

Forbus, K., Gentner, D., Everett, J. and Wu, M. (1997). Towards a computational model of evaluating and using analogical inferences, In Proceedings of Cognitive Science Conference.

Forbus, K., Hinrichs, T., Klenk, M., Lovett, A., Paritosh, P. and Usher, J. (2006). The FIRE Manual, Fire Version 2.0, Manual Version 1.0, Unpublished Manuscript as of 11/21/2006.

Forbus, K., Mostek, T., and Ferguson, R. 2002. An analogy ontology for integrating analogical processing and first principles reasoning. In Proceedings of IAAI-02, July 2002.

Forbus, K., Tomai, E., and Usher, J. 2003. Qualitative spatial reasoning for visual grouping in sketches. In Proceedings of the 16th International Workshop on Qualitative Reasoning, Brasilia.

Frank, G.; Farquhar, A.; & Fikes, R. Building a Large Knowledge Base from a Structured Source: The CIA World Fact Book. Knowledge Systems Laboratory, 1998.

Fried, L. S., and Holoyak, K. J. (1984). Induction of Category Distributions: A Framework for Classification Learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10, 234-257.

Friedland, N. and Allen, P., 2004, The Halo Pilot: Towards a Digital Aristotle, manuscript available at http://www.projecthalo.com/content/docs/halopilot_vulcan_finalreport.pdf

Gelman, R., and Gallistel, C. R. (1978). *The child's understanding of number*. Cambridge, MA: Harvard Univ. Press.

Gallistel, C. R., and Gelman, R. (2000) Nonverbal numerical cognition: from reals to integers. *Trends Cogn. Sci.* 4.

Gallistel, C.R. and Gelman, R. (1992) Preverbal and verbal counting and computation. *Cognition* 44, 43–74.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155-170.

Gigerenzer, G., Todd, P. M. & the ABC Research Group. (1999). *Simple heuristics that make us smart*. New York: Oxford University Press.

Goldstone, R. L. and Rogosky, B. J., (2002). Using relations within conceptual systems to translate across conceptual systems, *Cognition*, 84, 295-320.

Goodman, N., 1955. *Fact, fiction, and forecast*. Cambridge, MA: Harvard University Press.

Greeno, J. G. (1991). Number sense as situated knowing in a conceptual domain. *Journal of Research in Mathematics Education*, 22, 170–218.

- Griffiths, T. L., & Tenenbaum, J. B. (2006). Optimal predictions in everyday cognition. *Psychological Science*, 17, 767-773.
- Guerrin, F. (1995). Dualistic algebra for qualitative analysis. In Proceedings of the 9th international workshop on Qualitative Reasoning, Amsterdam, Holland.
- Hadjichristidis, C., Sloman, S.A., Stevenson, R.J., Over, D.E. 2004. Feature centrality and property induction. *Cognitive Science*, 28, 45-74.
- Halstead, D. and Forbus, K. 2005. Transforming between Propositions and Features: Bridging the Gap. Proceedings of AAAI-2005. Pittsburgh, PA.
- Hammond, K.R., McClelland, G.H. & Mumpower, J. (1980). *Human Judgement and Decision Making*. New York: Praeger.
- Harabagiu, S. and Lacatusu, F., 2004, Strategies for Advanced Question Answering, Proceedings of the Work-shop on Pragmatics of Question Answering at HLT-NAACL 2004.
- Harnad, S. (1987). *Categorical perception*. Cambridge: Cambridge University Press.
- Harte, J. (1988). *Consider a spherical cow: A course in environmental problem solving*, University Science Books, Sausalito, CA.
- Harte, J. (2001). *Consider a Cylindrical Cow: More Adventures in Environmental Problem Solving*. University Science Books, Sausalito, California.
- Higgins, E.T. (1996). Knowledge Activation: Accessibility, applicability, and salience. In E.T. Higgins and A.W.Kruglanski (Eds.), *Social Psychology: Handbook of basic principles* (pp239-270). New York: The Guilford Press.
- Henderson, P.W. and Peterson, R.A., 1992, Mental Accounting and Categorization, *Organizational Behavior and Human Decision Processes*, 51, 92-117.
- Holoyak, K. J., and Mah, W. A. (1984). Cognitive Reference Points in Judgments of Symbolic Magnitude. *Cognitive Psychology*, 14, 328-352.
- Holyoak, K. J. and Thagard, P. R. (1989). Analogical Mapping by Constraint Satisfaction, *Cognitive Science*, 13, 295-355.
- Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C., 2001, Question Answering in Webclopedia, Proceedings of the TREC-9 Conference.

Hummel, J.E and Holyoak, K. J. (1997). Distributed representations of structure: a theory of analogical access and mapping, *Psychological Review*, 104, 427-466.

Joram, E., Subrahmanyam, K., & Gelman, R. (1998). Measurement estimation: Learning to map the route from number to quantity and back. *Review of Educational Research*, Winter 68(4), 413-449.

Kahneman, D., and Tversky, A. Subjective probability: A judgment of representativeness. *Cognitive Psychology*, 1972, 3, 430-454.

Kahneman, D., and Tversky, A. 1979. Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, 47, 263-292.

Kahneman, D. (1992). Reference points, anchors, norms, and mixed feelings. *Organizational behavior and Human Decision Processes*, 51, 296-312.

Kammen, D. and Hassenzahl, D. (1999). *Should We Risk It? Exploring Environmental, Health and Technological Problem Solving*, Princeton University Press, Princeton, New Jersey.

Kareev, Y., Lieberman, I., and Lev, M. (1997). Through a Narrow Window: Sample Size and Perception of Correlation, *Journal of Experimental Psychology: General*, 126, 3, 278-287.

Katz, B., Borchardt, G. and Felshin, S., 2005a, Syntactic and Semantic Decomposition Strategies for Question Answering from Multiple Sources. *Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering*, 35-41, Pittsburgh, PA.

Katz, B., Marton, G., Borchardt, G., Brownell, A., Felshin, S., Loreto, D., Louis-Rosenberg, J., Lu, B., Mora, F., Stiller, S., Uzuner, O., and Wilcox, 2005b, A.. External Knowledge Sources for Question Answering *Proceedings of the 14th Annual Text REtrieval Conference*, Gaithersburg, MD.

Kemler, D.G. (1982). The ability for dimensional analysis in preschool and retarded children: Evidence from comparison, conservation, and prediction tasks. *Journal of Experimental Child Psychology*, 34, 469-489.

Klein, G. (1999). *Sources of Power: How People Make Decisions*. MIT Press.

Klenk, M. and Forbus, K. (2007). Cross domain analogies for learning domain theories. In Angela Scherwing et al. (Eds.), *Analogies: Integrating Multiple Cognitive Abilities*. Publications of the Institute of Cognitive Science, University of Osnabrück, Volume 5, 2007.

Kennedy, C. (2003). *Towards a Grammar of Vagueness*. Presented at the Princeton Semantics Workshop, May 17, 2003

- Kraus, S., Ryan, C. S., Judd, C. M., Hastie R., and Park, B. (1993). Use of mental frequency distributions to represent variability among members of social categories. *Social Cognition*, 11(1), 22-43.
- Krifka, Manfred 1989. Nominal reference, temporal constitution and quantification in event semantics. In Renate Bartsch, Johan van Benthem and Peter van Emde Boas (eds.), *Semantics and Contextual Expressions* 75-115. Dordrecht: Foris.
- Kuehne, S. E., 2004, On the Representation of Physical Quantities in Natural Language Text, In *Proceedings of the 26th Cognitive Science Conference*, Chicago.
- Kuehne, S., Forbus, K., Gentner, D. and Quinn, B.(2000) SEQL: Category learning as progressive abstraction using structure mapping. *Proceedings of CogSci 2000*, August, 2000.
- Larkin, J. H., Frederick R., Carbonell, J. and Gugliotta, A. 1988. FERMI: A Flexible Expert Reasoner with Multi-Domain Inferencing, *Cognitive Science*, 12(1), 101-138.
- Leake, D. (Ed.) 1996. *Case-based Reasoning: Experiences, Lessons and Future Directions*, MIT Press.
- Lee, G., Seo, J., Lee, S., Jung, H., Cho, B., Lee, C., Kwak, B., Cha, J., Kim, D., An, J., Kim, H., 2001, SiteQ: Engineering High Performance QA system Using Lexico-Semantic Pattern Matching and Shallow NLP, *Proceedings of TREC 2001*.
- LeFevre, J. A., Greenham, S. L., & Waheed, N. (1993). The development of procedural and conceptual knowledge in computational estimation. *Cognition and Instruction*, 11, 95–132.
- Lenat, D.B. 1982. The nature of heuristics, *Artificial Intelligence*, Volume 19, Issue 2, Pages 189-249
- Lenat, D. B. and Guha, R. V. (1989). *Building large knowledge-based systems: Representation and inference in the Cyc project*, Addison-Wesley, Reading, MA.
- Lenat D.B., Prakash, M. and Sheperd M., 1986. *CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks*. *AI Magazine*.
- Lenhart K. Schubert and Matthew Tong, 2003, "Extracting and evaluating general world knowledge from the Brown Corpus", *Proceedings of the HLT-NAACL Workshop on Text Meaning*, Edmonton, Alberta, pp. 7-13.
- Lehnert, W. G., 1986, A conceptual theory of question answering. In B. J. Grosz, K. Sparck Jones, and B. L. Webber (Eds). *Natural Language Processing*, Kaufmann, Los Altos, CA, pp 651–657.

- Linder, B.M. (1991). Understanding estimation and its relation to engineering education, Ph.D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology.
- Madison, B.L. & Steen, L.A. (Ed.). (2003). Quantitative literacy: Why numeracy matters for schools and colleges. Princeton, NJ: The National Council on Education and the Disciplines.
- Malmi, R. A., and Samson, D.J. (1983). Intuitive Averaging of Categorized Numerical Stimuli, *Journal of Verbal Learning and Verbal Behavior*, 22, 547-559.
- Malt, B. and Smith, E. (1984). Correlated Properties in Natural Categories. *Journal of Verbal Learning and Verbal Behavior*, 23(2), 250-269.
- Markman, A. B., & Gentner, D. (1993). Structural alignment during similarity comparisons. *Cognitive Psychology*, 25, 431-467.
- Markman, A. B., & Gentner, D. (1996). Commonalities and differences in similarity comparisons. *Memory & Cognition*, 24(2), 235-249.
- Markman, A. B., and Wisniewski, E. J. 1997. Similar and Different: The Differentiation of Basic-Level Categories, *Journal of Experimental Psychology: Learning, Memory and Cognition*, 1997, 23(1), 54-70.
- Miller, G. A., 1995, Wordnet: A Lexical Database for English, *Communications of the ACM*, 38(11).
- Minton, S. 1988. Learning Search Control Knowledge. Kluwer Academic Publishers.
- Moldovan, D., Clark, C., Harabagiu, S., Maiorano, S., 2003, COGEX: A Logic Prover for Question Answering. In *Proceedings of HLT-NAACL 2003*.
- Morrison, P. 1963. Fermi Questions. *American Journal of Physics*, 31(8), 626-627.
- Mussweiler, T. and Strack, F. (2001). The semantics of anchoring. *Organizational Behavior and Human Decision Processes*, 86(2), 234-255.
- Narayan, S. Harabagiu, S., 2004, Question Answering Based on Semantic Structures, *Proceedings of the 20th International Conference on Computational Linguistics, Geneva*.
- Nayak, P.P. 1994. Causal Approximations, *Artificial Intelligence*, 70, 1-58.
- Newell, A. 1982. The Knowledge Level, *Artificial Intelligence*, 18, 87-127.

Newell, A. (1983) The heuristic of George Pólya and its relation to artificial intelligence. In R. Groner, M. Groner, M., & W. Bischof (Eds.), *Methods of heuristics* (pp. 195- 243). Hillsdale, NJ: Erlbaum.

Newell A. and Simon H. A.. 1963. GPS, a program that simulates human thought. In Edward A. Feigenbaum and Julian Feldman, editors, *Computers and Thought*, pages 279--296. McGraw-Hill, New York.

Nilsson, N. 1994. *Principles of Artificial Intelligence*, Morgan Kaufman.

Northcraft, G.B. and Neale, M.A. (1987). Experts, amateurs, and real estate: An anchoring-and-adjustment perspective on property pricing decisions. *Organizational Behavior and Human Decision Processes*, 39, 84-97.

Nyberg, E., Frederking, R., Mitamura, T., Bilotti, M., Han-nan, K., Hiyakumoto, L., Ko, J., Lin, F., Lita, L., Pedro, V., and Schlaikjer, A., 2005, JAVELIN I and II Systems at TREC 2005, *Proceedings of TREC 2005*.

O'Connor, M.P., and Spotila, J.M. (1992). Consider a spherical lizard: Animals, models and approximations, *American Zoologist*, 32, pp 179-193.

Packer, A. (2001). *What Mathematics Should Everyone Know and Be Able to Do?* Princeton, NJ: National Council on Education and the Disciplines.

Pantel, P., Ravichandran, D., and Hovy, E., 2004, *Towards Terascale Knowledge Acquisition*. In *Proceedings of Conference on Computational Linguistics*, pp. 771-777. Geneva, Switzerland.

Paritosh, P.K and Forbus, K.D. (2001). *Common Sense on the Envelope* , In *Proceedings of the 15th International Workshop in Qualitative Reasoning*, San Antonio, TX.

Paritosh, P.K. and Forbus, K.D. (2003). *Qualitative Modeling and Similarity in Back of the Envelope Reasoning*. In *Proceedings of the 25th Annual Conference of the Cognitive Society*, Boston.

Paritosh, P.K. (2003). *A Sketch of a Theory of Quantity*, In *Proceedings of the 17th International Workshop on Qualitative Reasoning*, Brasilia, Brazil.

Paritosh, P.K. and Forbus, K.D. (2004). *Using Strategies and AND/OR Decomposition for Back of the Envelope Reasoning*. In *Proceedings of the 18th International Workshop on Qualitative Reasoning*, Evanston.

Paritosh, P.K. (2004). *Symbolizing Quantity*. In *Proceedings of the 26th Cognitive Science Conference*, Chicago.

Paritosh, P.K. and Forbus, K.D., (2005). Analysis of Strategic Knowledge in Back of the Envelope Reasoning, In Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05), Pittsburgh, PA.

Paritosh, P.K. and Klenk, M.E. (2006). Cognitive Processes in Quantitative Estimation: Analogical Anchors and Causal Adjustment. In the Proceedings of the 28th Annual Conference of the Cognitive Science Society, Vancouver.

Paritosh, P.K. (2006). The Heuristic Reasoning Manifesto. In Proceedings of the 20th International Workshop on Qualitative Reasoning, Dartmouth.

Paritosh, P.K. (2007). Beyond Corpus Lookup: Towards Heuristic Reasoning with Text. To appear in the Proceedings of the 3rd International Workshop on Knowledge and Reasoning in Answering Questions, at IJCAI-07, Hyderabad.

Paulos, J.A. (1988). Innumeracy: Mathematical illiteracy and its consequences. New York: Hill and Wang.

Peterson, C.R., and Beach, L.R. (1967). Man as an intuitive statistician, *Psychological Bulletin*, 68(1), pp 29-46.

Pisan, Y., 1998. An integrated architecture for engineering problem solving. Doctoral dissertation, Northwestern University, Evanston, IL, USA.

Polya, G. 1945. *How to Solve It*, Princeton University Press, Princeton, NJ.

Porter, T. 1997. "The triumph of numbers: civic implications of quantitative literacy," in Lynn Steen, (ed.), *Why Numbers Count: Quantitative Literacy for Tomorrow's America* New York, 5-10. The College Entrance Examination Board.

Ram, A. and Santamaria, J.C. (1997). Continuous case-based reasoning. *Artificial Intelligence*, 90, 25-77

Rips, L. J., and Turbull, W. (1980) How big is big? Relative and absolute properties in memory. *Cognition*, 8, 145-174.

Rosch, E. (1975). Cognitive Reference Points. *Cognitive Psychology*, 7, 532-547.

Rosch, E., 1978. "Principles of categorization" in Rosch, E. and B.B. Lloyd, (eds.), *Cognition and Categorization*, Hillsdale, N.J.: Erlbaum.

- Ryalls, B. O. and Smith, L. B. (2000). Adults Acquisition of Novel Dimension Words: Creating a Semantic Congruity Effect, *Journal of General Psychology*, 127(3), 279-326.
- Sanghi, M., Paritosh P.K., and Thomas, R. (2005). Sub-linear Algorithms for Landmark Discovery from Black Box Models. In *Proceedings of the 19th International Workshop on Qualitative Reasoning*, Graz, Austria.
- Schlobach S., Ahn D., de Rijke M., and Jijkoun V., in press, Data-driven Type Checking in Open Domain Question Answering, *Journal of Applied Logic*.
- Schmidt-Nielsen K, Scaling. (1984). Why is animal size so important? Cambridge University Press.
- Schwarz, J. (1996). Semantic Aspects of Quantity. Unpublished Manuscript, Harvard University, retrieved from <http://gseweb.harvard.edu/~faculty/schwartz/semantic.htm>
- Shah, P., Schneider, D., Matuszek, C., Kahlert, R., Aldag, B., Baxter, D., Cabral, J., Witbrock, M., Curtis, P. 2006. Automated Population of Cyc: Extracting Information about Named-entities from the Web. In *Proceedings of the Nineteenth International FLAIRS Conference*.
- Shepard, R. N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function, I. *Psychometrika*, 27(2), 125-140.
- Simon, H. A. 1957. A Behavioral Model of Rational Choice, in *Models of Man*.
- Singh, Push, Lin, Thomas, Mueller, Erik T., Lim, Grace, Perkins, Travell, & Zhu, Wan Li (2002). Open Mind Common Sense: Knowledge acquisition from the general public. In Robert Meersman & Zahir Tari (Eds.), *Lecture Notes in Computer Science: Vol. 2519. On the Move to Meaningful Internet Systems 2002*, (pp. 1223-1237). Heidelberg: Springer-Verlag.
- Skorstad, J., Gentner, D., & Medin, D. (1988). Abstraction processes during concept learning: A structural view. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, 419-425.
- Slagle, J. 1963. A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus, *Journal of the ACM*, 10 (4), 507-520.
- Smith, L.B., 1984. Young children's understanding of attributes and dimensions: a comparison of conceptual and linguistic measures. *Child Development* 55, 363–380.
- Sowder, J. T. (1992). Estimation and number sense, in D. A. Grouws, (ed.), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Co., New York, pp. 371–389.

- Staab, S. and Hahn, U. (1998). Grading on the Fly. In Proceedings of the 20th Annual Meeting of the Cognitive Science Society, Madison, WI.
- Szirtes, T. and Rosza, P. 1998. Applied Dimensional Analysis and Modeling. McGraw Hill, New York.
- Swartz, C. E., 2003. Back-of-the-Envelope Physics. Johns Hopkins University Press, Maryland.
- Thaler, R.H., (1999). Mental accounting matters, *J. Behav. Dec. Making*, 12, 183-206.
- Thompson, C.P., Skowronski, J.J., Larsen, S.F., and Betz, A.L., 1996. Autobiographical Memory: Remembering What and Remembering When, Lawrence Erlbaum, NJ.
- Torgerson, W. S. (1965). Multidimensional scaling of similarity. *Psychometrika*, 30(4).
- Triplehorn, D. 1994-95. On the Back of an Envelope, *Journal of Geological Education*, 42, 43.
- Tversky, A. (1977). Features of Similarity, *Psychological Review* 84(4), pp 327 - 352.
- Tversky, A., & Kahneman, D. Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, 1973, 5, 207-232.
- Tversky, A., and Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases, *Science*, 185, pp 1124-1131.
- Varzi, A. C. (2003). Vagueness, In *Encyclopedia of Cognitive Science*, Macmillan and Nature Publishing Group, London.
- Weisskopf, V. 1984-86. Search for Simplicity, *American Journal of Physics*, 52, 53, 54.
- Weld, D.S. 1987. Comparative Analysis. In Proceedings of IJCAI, 959-965.
- Yan, J. and Forbus, K. 2004. Similarity-based qualitative simulation: A preliminary report. In Proceedings of the 18th International Qualitative Reasoning Workshop, Evanston.
- Zadeh, L. (1965). Fuzzy Sets, *Information and Control*, 8, 338-353.

Appendix A: Suggestions used by BotE-Solver

```

(defSuggestion HomogenousGroupExtensiveQuantityStrategy
  (valueOf ((QPQuantityFn ?quantity) ?object) ?ans)
  :documentation "If there is a group of n of something, and each one
    of them has a value q, then the value for the all of
    them is n*q if q is an extensive dimension"
  :test (and (isa ?quantity ExtensiveMeasurableQuantitySlot)
    (isa ?object Group))
  :subgoals ((groupMemberType ?object ?individual-member)
    (valueOf ((QPQuantityFn numberOfGroupMembers) ?object) ?count)
    (valueOf ((QPQuantityFn ?quantity) ?individual-member)
      ?individual-value))
  :result-step (evaluate ?ans (TimesFn ?count ?individual-value)))

(defSuggestion CountGroupMembersByMereologyStrategy
  (valueOf ((QPQuantityFn numberOfGroupMembers) ?group) ?ans)
  :documentation "Add up counts from subgroups to produce a total"
  :test (isa ?group Group)
  :subgoals ((evaluate ?subgroups
    (TheClosedRetrievalSetOf
      ?subgroup (subGroups ?group ?subgroup)))
    (evaluate ?subgroup-counts
      (MapFunctionOverList
        (FunctionToArg 2
          (Kappa (?s)
            (valueOf
              ((QPQuantityFn
                numberOfGroupMembers) ?s) ?val)))
        (SetToListFn ?subgroups))))
  :result-step (evaluate ?ans (PlusAll IdentityFn ?subgroup-counts)))

(defSuggestion BalanceStrategy
  (valueOf ((QPQuantityFn ?quantity) ?system ?state) ?val)
  :test (and (isa ?quantity ConservedQuantity)
    (isa ?system System)
    (quantityConservedForSystem ?quantity ?system))
  :subgoals ((and (stateOfSystem ?system ?alt-state)
    (different ?state ?alt-state))
    (valueOf ((QPQuantityFn ?quantity) ?system ?alt-state) ?alt-val))
  :result-step (evaluate ?val (IdentityFn ?alt-val)))

(defSuggestion TotalEnergyConversionStrategy
  (valueOf ((QPQuantityFn energyProduced) ?event) ?energy)
  :documentation "E=mc^2"
  :test (isa ?event TotalEnergyConversionProcess)
  :subgoals ((objectActedOn ?event ?obj)
    (valueOf ((QPQuantityFn massOfObject) ?obj) ?mass)
    (valueOf ((QPQuantityFn velocityOfObject) Light) ?c))
  :result-step (evaluate ?energy (TimesFn ?mass (ExponentFn ?c 2))))

```



```

(defSuggestion DensityStrategy
  (valueOf ((QPQuantityFn ?quantity) ?object) ?ans)
  :documentation "implements density heuristic"
  :subgoals ((and (densityQuantityFor ?quantity ?density-quantity)
                  (extentQuantityFor ?density-quantity ?extent-quantity)
                  (valueOf ((QPQuantityFn ?density-quantity) ?object) ?density)
                  (valueOf ((QPQuantityFn ?extent-quantity) ?object) ?extent))
             :result-step (evaluate ?ans (TimesFn ?density ?extent)))

(defSuggestion AmountViaLatentHeatOfVaporizationStrategy
  (valueOf ((QPQuantityFn amountOfParticipant) ?event ?object) ?ans)
  :test (and (isa ?event HeatingProcess)
             (objectActedOn ?event ?object))
  :documentation "Finds amount of a substance in a heating process"
  :subgoals ((isa ?object ChemicalSubstanceType)
             (valueOf ((QPQuantityFn amountOfEnergyUsed) ?event)
                  ?heat-amount)
             (valueOf ((QPQuantityFn heatOfVaporization) ?substance)
                  ?latent-heat))
  :result-step (evaluate ?ans (QuotientFn ?heat-amount ?latent-heat)))

(defSuggestion ChangeOfQuantityInEventStrategy
  (valueOf ((QPQuantityFn changeInQuantity) ?quantity ?event) ?ans)
  :documentation "compute change in a quantity by subtracting
                 initial from final"
  :test (isa ?event Event)
  :subgoals ((valueOf ((QPQuantityFn initialValueOfQuantity)
                      ?quantity ?event) ?initial-value)
             (valueOf ((QPQuantityFn finalValueOfQuantity)
                      ?quantity ?event) ?final-value))
  :result-step (evaluate ?ans (MinusFn ?final-value ?initial-value)))

(defSuggestion TempChangeInFallStrategy
  (valueOf ((QPQuantityFn changeInQuantity)
           temperatureOfObject ?event) ?delta-t)
  :documentation "Compute change in temperature if all potential
                 energy were converted into heat via delta-t=g.h/c"
  :test (or (isa ?event HeatingProcess)
            (isa ?event CoolingProcess))
  :subgoals ((objectActedOn ?event ?object)
             (isa ?object ChemicalSubstanceType)
             (valueOf ((QPQuantityFn specificHeatCapacity) ?object) ?c)
             (valueOf ((QPQuantityFn verticalFallDistance)
                      ?object ?event) ?h)
             (valueOf Gravity-UnitOfAcceleration ?g))
  :result-step (evaluate ?delta-t (QuotientFn (TimesFn ?g ?h) ?c)))

```



```

(defSuggestion IntensiveQuantityViaHomogenousMereology
  (valueOf ((QPQuantityFn ?quantity) ?object) ?ans)
  :test (and (isa ?quantity IntensiveMeasurableQuantitySlot)
             (isa ?object HomogeneousStructure))
  :subgoals ((constituents ?object ?constituent)
             (valueOf ((QPQuantityFn ?quantity) ?constituent) ?ans)))

(defSuggestion VolumeViaCrossSection
  (valueOf ((QPQuantityFn volumeOfObject) ?object) ?ans)
  :test (isa ?object ThreeDimensionalThing)
  :subgoals ((valueOf ((QPQuantityFn areaOfObject) ?object) ?area)
             (valueOf ((QPQuantityFn heightOfObject) ?object) ?height))
  :result-step (evaluate ?ans (TimesFn ?area ?height)))

(defSuggestion SpatialSurfaceAligned
  (valueOf ((QPQuantityFn areaOfObject) ?object) ?ans)
  :subgoals ((alignedAlongSurface ?object ?alt-object)
             (valueOf ((QPQuantityFn surfaceAreaOfTopOfObject) ?alt-
object) ?ans)))

(defSuggestion SpatialLengthAligned
  (valueOf ((QPQuantityFn ?quantity) ?object) ?ans)
  :test (isa ?quantity DistanceQuantitySlot)
  :subgoals ((alignedAlongQuantity ?object ?alt-object ?quantity)
             (valueOf ((QPQuantityFn ?quantity) ?alt-object) ?ans)))

(defSuggestion ComputeRectangularArea
  (valueOf ((QPQuantityFn areaOfObject) ?object) ?ans)
  :subgoals ((valueOf ((QPQuantityFn lengthOfObject) ?object) ?length)
             (valueOf ((QPQuantityFn widthOfObject) ?object) ?width))
  :result-step (evaluate ?ans (TimesFn ?length ?width)))

(defSuggestion EnergyByUsageStrategy
  (valueOf ((QPQuantityFn ?quantity) ?object) ?ans)
  :test (isa ?quantity EnergyQuantitySlot)
  :subgoals ( ;; we know of an event where the object is the energy source
             (energySource ?event ?object)
             (valueOf ((QPQuantityFn energyProduced) ?event) ?energy-out)
             (objectActedOn ?event ?system)
             (amountOfStuffInRole ?event ?stuff ?amount energySource)
             (valueOf ((QPQuantityFn levelOfEfficiency) ?system) ?
efficiency))
  :result-step (evaluate ?ans (QuotientFn ?energy-out ?efficiency)))

(defSuggestion EnergyViaMotionStrategy
  (valueOf ((QPQuantityFn energyProduced) ?event) ?ans)
  :documentation "In a translation event Energy/Work = Force x Distance"
  :test (isa ?event Movement-TranslationEvent)
  :subgoals ((objectActedOn ?event ?object)

```

```

                (valueOf ((QPQuantityFn forceActingOnObject) ?object) ?force)
                (valueOf ((QPQuantityFn distanceTranslated) ?event) ?
distance))
        :result-step (evaluate ?ans (TimesFn ?force ?distance)))

(defSuggestion ForceAtConstantVelocityStrategy
  (valueOf ((QPQuantityFn forceActingOnObject) ?object) ?force)
  :test (valueOf ((QPQuantityFn accelerationOfObject-Translation) ?object)
0)
  :subgoals ((objectActedOn ?event ?object)
              (valueOf ((QPQuantityFn frictionalForce) ?event) ?friction))
  :result-step (evaluate ?force (IdentityFn ?friction)))

(defSuggestion FrictionalForceStrategy
  (valueOf ((QPQuantityFn frictionalForce) ?event) ?force)
  :test (and (isa ?event Movement-TranslationEvent)
              (directionOfTranslation-Throughout ?event Horizontal-
Generally))
  :subgoals ((objectActedOn ?event ?object)
              (movementSurface ?event ?surface)
              (valueOf ((QPQuantityFn massOfObject) ?object) ?mass)
              (valueOf ((QPQuantityFn coefficientOfFriction) ?object ?
surface) ?coeff)
              (valueOf Gravity-UnitOfAcceleration ?g))
  :result-step (evaluate ?force (TimesFn ?coeff ?mass ?g)))

(defSuggestion NewtonsLawOfGravitation
  (valueOf ((QPQuantityFn gravitationalForceBetween) ?object1 ?
object2) ?ans)
  :subgoals ((valueOf ((QPQuantityFn massOfObject) ?object1) ?mass1)
              (valueOf ((QPQuantityFn massOfObject) ?object2) ?mass2)
              (valueOf ((QPQuantityFn distanceBetween) ?object1 ?object2) ?
distance))
  :result-step (evaluate ?ans (QuotientFn (TimesFn 6.67e-11 ?mass1 ?mass2)
                                           (ExponentFn ?distance 2))))

(defSuggestion RadiusViaCentrifugalAcceleration
  (valueOf ((QPQuantityFn radiusOfTurn) ?event) ?ans)
  :test (isa ?event MovementEvent)
  :subgoals ((objectActedOn ?event ?object)
              (valueOf ((QPQuantityFn massOfObject) ?object) ?mass)
              (valueOf ((QPQuantityFn velocityOfObject) ?object) ?velocity)
              (valueOf ((QPQuantityFn accelerationOfObjectDuring) ?event) ?
acceleration))
  :result-step (evaluate ?ans (QuotientFn (TimesFn ?mass (ExponentFn ?
velocity 2)) ?acceleration)))

(defSuggestion CountViaConstituentStrategy
  (valueOf ((QPQuantityFn ?count-quantity) ?whole ?part) ?count)
  :documentation "If ?whole is made out of ?part, then the quantities that
describes

```

```

the dimension of the constitution can be used to measure
the count"
  :test (and (constituents ?whole ?part)
             (isa ?count-quantity CountQuantitySlot))
  :subgoals ((constituentPhysicalQuantity ?whole ?part ?constituent-
quantity)
             (valueOf ((QPQuantityFn ?constituent-quantity) ?whole) ?
whole-measure)
             (valueOf ((QPQuantityFn ?constituent-quantity) ?part) ?part-
measure))
  :result-step (evaluate ?count (QuotientFn ?whole-measure ?part-measure)))

(defSuggestion CylindricalVolume
  (valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)
  :documentation "Consider a cylindrical cow for computing volume"
  :test (and (isa ?object ThreeDimensionalThing)
             (uninferredSentence (shapeOfObject ?object ?shape)))
  :subgoals ((valueOf ((QPQuantityFn lengthOfObject) ?object) ?length)
             (valueOf ((QPQuantityFn widthOfObject) ?object) ?width))
  :result-step (evaluate ?volume (TimesFn 3 ?length (ExponentFn
(QuotientFn ?width 2) 2))))

(defSuggestion RingVolume
  (valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)
  :documentation "Volume of a Ring"
  :test (shapeOfObject ?object RingShapedObject)
  :subgoals ((valueOf ((QPQuantityFn innerRadius) ?object) ?radius-in)
             (valueOf ((QPQuantityFn outerRadius) ?object) ?radius-out)
             (valueOf ((QPQuantityFn widthOfObject) ?object) ?width))
  :result-step (evaluate ?volume
                (TimesFn 3.14
                        ?width
                        (DifferenceFn
                          (ExponentFn ?radius-out 2)
                          (ExponentFn ?radius-in 2)))))

(defSuggestion SphericalVolume
  (valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)
  :documentation "Consider a spherical cow for computing volume"
  :test (and (isa ?object ThreeDimensionalThing)
             (uninferredSentence (shapeOfObject ?object ?shape)))
  :subgoals ((valueOf ((QPQuantityFn extensionParametersOfObject) ?object)
?size))
  :result-step (evaluate ?volume (TimesFn 12 (ExponentFn (QuotientFn ?size
2) 3))))

(defSuggestion RectangularVolume
  (valueOf ((QPQuantityFn volumeOfObject) ?object) ?volume)
  :documentation "Volume of a rectangular parallelepiped"
  :test (shapeOfObject ?object RectangularParallelepiped)

```

```

:subgoals ((valueOf ((QPQuantityFn lengthOfObject) ?object) ?length)
           (valueOf ((QPQuantityFn widthOfObject) ?object) ?width)
           (valueOf ((QPQuantityFn heightOfObject) ?object) ?height))
:result-step (evaluate ?volume (TimesFn ?length ?width ?height))

(defSuggestion CountViaSignificantConstituentStrategy
  (valueOf ((QPQuantityFn ?count-quantity) ?whole ?constituent) ?count)
  :documentation "If ?whole is made out of ?main-part and we can find the
amount of
                ?constituent in just that part, then that is a good
estimate for
                the ?whole"
  :test (and (constituents ?whole ?constituent)
             (significantConstituentPart ?whole ?main-part ?constituent)
             (isa ?count-quantity CountQuantitySlot))
  :subgoals ((valueOf ((QPQuantityFn ?count-quantity) ?main-part ?
constituent) ?count)))

(defSuggestion ComputeTotalValueStrategy
  (valueOf ((QPQuantityFn totalValue) ?quantity ?object ?container) ?ans)
  :test (isa ?quantity ExtensiveMeasurableQuantitySlot)
  :subgoals ((valueOf ((QPQuantityFn ?quantity) ?object) ?ans-per-object)
             (valueOf ((QPQuantityFn countContained) ?object ?container) ?
count))
  :result-step (evaluate ?ans (TimesFn ?ans-per-object ?count)))

(defSuggestion CountViaContainmentStrategy
  (valueOf ((QPQuantityFn countContained) ?object ?container) ?count)
  :subgoals ((constituentPhysicalQuantity ?object ?container ?quantity)
             (valueOf ((QPQuantityFn ?quantity) ?object) ?ans-object)
             (valueOf ((QPQuantityFn ?quantity) ?container) ?ans-
container))
  :result-step (evaluate ?count (QuotientFn ?ans-container ?ans-object)))

(defSuggestion CountViaMereologyStrategy
  (valueOf ((QPQuantityFn countContained) ?contained ?container) ?ans)
  :subgoals ((physicalParts ?object ?contained)
             (valueOf ((QPQuantityFn countContained) ?contained ?object) ?
parts-per-object)
             (valueOf ((QPQuantityFn countContained) ?object ?container) ?
objects-per-container))
  :result-step (evaluate ?ans (TimesFn ?parts-per-object ?objects-per-
container)))

(defSuggestion CountDiscardedViaLifetime
  (valueOf ((QPQuantityFn countDiscardedPerYear) ?object ?where) ?ans)
  :subgoals ((valueOf ((QPQuantityFn countContained) ?object ?where) ?
total)
             (valueOf ((QPQuantityFn lifetimeOf) ?object) ?life))
  :result-step (evaluate ?ans (QuotientFn ?total ?life)))

(defSuggestion HouseholdStrategyForCounting

```

```

      (valueOf ((QPQuantityFn countContained) ?object ?place) ?count)
      :test (ownsObjectType Household ?object)
      :subgoals ((valueOf ((QPQuantityFn countContained) Household ?place) ?
num-households)
                 (valueOf ((QPQuantityFn countContained) ?object Household) ?
num-per-household))
      :result-step (evaluate ?count (TimesFn ?num-households ?num-per-
household)))

(defSuggestion CountGroupsViaSize
  (valueOf ((QPQuantityFn countContained) ?contained ?container) ?ans)
  :test (and (isa ?contained Group)
             (isa ?container Group))
  :subgoals ((valueOf ((QPQuantityFn numberOfGroupMembers) ?contained) ?
members)
             (valueOf ((QPQuantityFn populationOf) ?container) ?
population))
  :result-step (evaluate ?ans (QuotientFn ?population ?members)))

(defSuggestion CountViaEventStrategy
  (valueOf ((QPQuantityFn countContained) ?agent ?place) ?count)
  :test (isa ?agent Agent-Generic)
  :subgoals ((performedBy ?agent ?event)
             (valueOf ((QPQuantityFn countContained) ?event ?place) ?num-
events)
             (valueOf ((QPQuantityFn productionCapacity) ?agent ?event) ?
num-acts))
  :result-step (evaluate ?count (QuotientFn ?num-events ?num-acts)))

(defSuggestion AgentCapacityStrategy
  (valueOf ((QPQuantityFn productionCapacity) ?agent ?event) ?count)
  :subgoals ((valueOf ((QPQuantityFn duration) ?event) ?duration)
             (valueOf ((QPQuantityFn timeWorkedOnTask) ?agent ?event) ?
total-time))
  :result-step (evaluate ?count (QuotientFn ?total-time ?duration)))

(defSuggestion CountEventsByDeviceStrategy
  (valueOf ((QPQuantityFn countContained) ?event ?place) ?num-events)
  :test (isa ?event Event)
  :subgoals ((deviceUsed ?event ?device)
             (valueOf ((QPQuantityFn countContained) ?device ?place) ?num-
devices)
             (valueOf ((QPQuantityFn frequencyOfEvent) ?event) ?freq))
  :result-step (evaluate ?num-events (QuotientFn ?num-devices ?freq)))

(defSuggestion EnergyBalanceViaPowerRatingStrategy
  (valueOf ((QPQuantityFn energyProduced) ?event) ?ans)
  :subgoals ((energySource ?event ?source)
             (valueOf ((QPQuantityFn powerRating) ?source) ?consumed)
             (objectActedOn ?event ?device))

```

```

                (valueOf ((QPQuantityFn levelOfEfficiency) ?device) ?
efficiency))
        :result-step (evaluate ?ans (TimesFn ?consumed ?efficiency)))

(defSuggestion EnergyViaElectricPotential
  (valueOf ((QPQuantityFn energyProduced) ?event) ?ans)
  :test (isa ?event ElectricalProcess)
  :subgoals ((objectActedOn ?event ?object)
             (valueOf ((QPQuantityFn chargeOfObject) ?object) ?charge)
             (valueOf ((QPQuantityFn electricalPotentialDifference) ?
event) ?potential))
  :result-step (evaluate ?ans (TimesFn ?charge ?potential)))

(defSuggestion ComputeSurfaceAreaOfWholeObject
  (valueOf ((QPQuantityFn surfaceAreaOfWholeObject) ?object) ?ans)
  :test (isa ?object ThreeDimensionalThing)
  :subgoals ((valueOf ((QPQuantityFn lengthOfObject) ?object) ?length)
             (valueOf ((QPQuantityFn heightOfObject) ?object) ?height)
             (valueOf ((QPQuantityFn widthOfObject) ?object) ?width))
  :result-step (evaluate ?ans (TimesFn 2 (PlusFn (TimesFn ?length ?height)
                                                (TimesFn ?length ?width)
                                                (TimesFn ?width ?
height))))))

(defSuggestion AmountViaUniformDistribution
  (valueOf ((QPQuantityFn ?quantity) ?object ?event) ?ans)
  :test (and (isa ?event UniformDistributionEvent)
            (isa ?quantity ExtensiveMeasurableQuantitySlot))
  :subgoals ((valueOf ((QPQuantityFn totalValue) ?quantity ?object ?event)
?total)
            (valueOf ((QPQuantityFn countContained) ?object ?event) ?
count))
  :result-step (evaluate ?ans (QuotientFn ?total ?count)))

(defSuggestion CountViaLocation
  (valueOf ((QPQuantityFn countContained) ?object ?event) ?ans)
  :test (isa ?event Event-Localized)
  :subgoals ((eventOccursAtLocation ?event ?location)
            (valueOf ((QPQuantityFn countContained) ?object ?location) ?
count)
            (valueOf ((QPQuantityFn fractionParticipating) ?event ?
location) ?fraction))
  :result-step (evaluate ?ans (TimesFn ?count ?fraction)))

(defSuggestion CountViaPopulation
  (valueOf ((QPQuantityFn countContained) ?object ?location) ?ans)
  :test (isa ?location GeographicalRegion)
  :subgoals ((valueOf ((QPQuantityFn populationOf) ?location) ?ans)))

(defSuggestion DurationViaRateStrategy
  (valueOf ((QPQuantityFn duration) ?event) ?ans)

```



```

:test (isa ?event Event)
:subgoals ((performedBy ?event ?performer)
           (rateQuantityFor ?event ?rate-quantity)
           (amountQuantityFor ?event ?amount-quantity)
           (valueOf ((QPQuantityFn ?amount-quantity) ?event) ?amount)
           (valueOf ((QPQuantityFn ?rate-quantity) ?performer) ?rate))
:result-step (evaluate ?ans (QuotientFn ?amount ?rate)))

(defSuggestion ComputeDistance
  (valueOf ((QPQuantityFn distanceTranslated) ?event) ?ans)
:test (isa ?event MovementEvent)
:subgoals ((fromLocation ?event ?start)
           (toLocation ?event ?end)
           (valueOf ((QPQuantityFn distanceBetween) ?start ?end) ?ans)))

(defSuggestion ScaleModelQuantityStrategy
  (valueOf ((QPQuantityFn ?quantity)
           (ScaleModelAnalogueFn ?model ?base-object-1)
           (ScaleModelAnalogueFn ?model ?base-object-2))
           ?ans)
:test (isa ?model ScaleModel)
:subgoals ((valueOf ((QPQuantityFn ?quantity) ?base-object-1 ?base-
object-2) ?base-value)
           (valueOf ((QPQuantityFn scalingFactor) ?model) ?scaling-
factor))
:result-step (evaluate ?ans (TimesFn ?scaling-factor ?base-value)))

(defSuggestion ScalingFactorViaRatioStrategy
  (valueOf ((QPQuantityFn scalingFactor) ?model) ?ans)
:test (isa ?model ScaleModel)
:subgoals ((scaledObject ?model ?base-object)
           (correspondsInMapping ?model ?base-object ?target-object)
           (scaleModelQuantity ?model ?quantity)
           (valueOf ((QPQuantityFn ?quantity) ?base-object) ?base-val)
           (valueOf ((QPQuantityFn ?quantity) ?target-object) ?target-
val))
:result-step (evaluate ?ans (QuotientFn ?target-val ?base-val)))

```

Appendix B: Sample Case from Basketball Domain

```

(ist-Information JasonKiddDescription-Enriched
  (isa JasonKidd
    (LowValueContextualizedFn heightOfObject
      BasketballPlayers)))
(ist-Information JasonKiddDescription-Enriched
  (isa JasonKidd
    (LowValueContextualizedFn seasonFreeThrowPercent
      BasketballPlayers)))
(ist-Information JasonKiddDescription-Enriched
  (isa JasonKidd
    (HighValueContextualizedFn seasonThreePointsPercent
      BasketballPlayers)))
(ist-Information JasonKiddDescription-Enriched
  (isa JasonKidd
    (MediumValueContextualizedFn seasonReboundsPerGame
      BasketballPlayers)))
(ist-Information JasonKiddDescription-Enriched
  (isa JasonKidd
    (MediumValueContextualizedFn seasonAssistsPerGame
      BasketballPlayers)))
(ist-Information JasonKiddDescription-Enriched
  (isa JasonKidd
    (MediumValueContextualizedFn seasonPointsPerGame
      BasketballPlayers)))
(isa JasonKiddDescription-Enriched Case)
(ist-Information JasonKiddDescription-Enriched
  (isa JasonKidd BasketballPointGuard))
(ist-Information JasonKiddDescription-Enriched
  (seasonPointsPerGame JasonKidd 13.5))
(ist-Information JasonKiddDescription-Enriched
  (seasonAssistsPerGame JasonKidd 7.1))
(ist-Information JasonKiddDescription-Enriched
  (seasonReboundsPerGame JasonKidd 6.8))
(ist-Information JasonKiddDescription-Enriched
  (seasonThreePointsPercent JasonKidd 0.404))
(ist-Information JasonKiddDescription-Enriched
  (seasonFreeThrowPercent JasonKidd 0.718))
(ist-Information JasonKiddDescription-Enriched
  (heightOfObject JasonKidd 1.93))
(ist-Information JasonKiddDescription-Enriched
  (qprop heightOfObject seasonReboundsPerGame
    BasketballPlayers))
(ist-Information JasonKiddDescription-Enriched
  (qprop seasonThreePointPercent seasonFreeThrowPercent
    BasketballPlayers))
(elementOf JasonKiddDescription-Enriched
  (CaseLibraryFn BasketballPlayers-Enriched))

```