Beyond Corpus Lookup: Towards Heuristic Reasoning with Text

Praveen Paritosh
Qualitative Reasoning Group, Northwestern University
2145 Sheridan Road, Evanston IL 60208.
paritosh@cs.northwestern.edu

Abstract

Both knowledge-based and text-based approaches to question answering suffer from brittleness. Text-based approaches use existing corpora like the web to answer a broad array of questions. However, most of these systems rely on corpus lookup. They have limited abilities to combine information from multiple sources, draw inferences, double check and explain answers. Knowledge-based systems can make sophisticated inferences, however, currently there exists no knowledge base that is broad enough to perform reasonably in the TREC question answering competitions. We argue that both these approaches can leverage off each other at *heuristic reasoning*, which is the type of reasoning underlying the human ability to make educated guesses. We focus on back of the envelope reasoning, the process of generating ballpark quantitative estimates. Starting from our previous work on a knowledge-based back of the envelope problem solver [Paritosh and Forbus, 2005], we present an analysis of how back of the envelope reasoning can be done with text.

1 Introduction

Answering questions is a common task for many different paradigms of AI research. In the knowledge-based (KB) approach, it is typically addressed as *problem solving*, and is accomplished by reasoning with formal representations provided with the problem and/or available in a knowledge base. In the text-based (TB) approach, it is typically called *question answering* (QA), and is accomplished by retrieving and analyzing relevant text documents from a corpus.

Both of these approaches have strengths and weaknesses. The KB 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 evaluation[1], 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 TB 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 TB 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 mother consume per day?" Simple chains of reasoning can reject this answer, but most TB approaches will find this difficult.

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, the ability to flexibly generate educated guesses even when the "correct" answers are much harder to find. This paper focuses on back of the envelope reasoning, e.g., generating rough estimates to questions like "How much money is spent on healthcare in the US?" as a domain for heuristic reasoning.

In the next section we look at two architectures for integrating text and knowledge-based systems. In Section 3, we present an analysis of brittleness in text and knowledge and describe heuristic reasoning. Section 4 describes the formal representations for back of the envelope reasoning. Section 5 and 6 present an analysis of textual expressions of questions and heuristic methods in back of the envelope reasoning.

2 Integrating Text and Knowledge

In this section, we describe two different architectures for integrating knowledge and text based approaches. The *knowledge-as-driver* approach consists of a knowledge based system which uses text corpora (e.g., web) as an auxilliary resource, possibly, when its knowledge base fails. Figure 1 shows a schematic of this approach. The *text reformulator* takes a query in formal representation, e.g., predicate calculus, and converts it into text queries which then uses the full power of the text based system to come back with an answer. The answer string is mapped on to existing concept in its knowledge, or if no such concept

[1] http://www.projecthalo.com/

exists in the knowledge base, a new concept is created by the *knowledge reformulator*. This is the approach taken by Shah et al. [2006] to build a system that extends the knowledge in the Cyc KB from the web. Such a system cannot reason about arbitrary knowledge that is available in the text corpus. If the knowledge base already contains concepts of population, country, and number; then it can be used to gather populations of countries. However if the concept of capitals of countries is not in the knowledge base, then it will not be able to answer and reason about queries regarding capitals. More work needs to be done to push on the breadth of knowledge as text that is usable by such a system. There are considerable challenges in building a knowledge reformulator that can robustly handle the noise and variability in text.

The *text-as-driver* approach consists of a text based system that uses knowledge to refine, verify [Schlobach et al., in press], and augment its results. There are many systems that use ontologies, e.g., Wordnet [Miller, 1995], for query expansion [e.g., Hovy et al, 2001]. There are also attempts to learn such conceptual relations from text, e.g.,

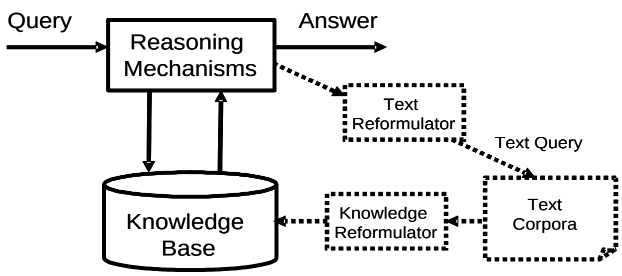

Figure 1: A knowledge-as-driver approach to integrating knowledge and text. The parts show in solid are a simplified picture of the knowledge-based component, and the dotted is the text-based component.

using Wikipedia [Katz et al., 2005; see also Pantel et al, 2004]. The JAVELIN QA system [Nyberg et al., 2005] uses a planner to control the information flow. The COGEX system [Moldovan et al., 2003] uses a theorem prover which uses world knowledge in the form of axioms extracted from Wordnet glosses. They report an impressive 30% improvement in performance as a result of the theorem prover. Such a system can reason about arbitrary knowledge that is available in the text corpus, however, the reasoning is limited. Much of the reasoning is exploiting type information and *isa* hierarchies. Another line of reasoning with text is the PASCAL Recognizing Textual Entailment (RTE) task. The goal of the task is to recognize, given two text fragments, whether the meaning of one text fragment can be inferred (entailed) from the other [Dagan et al., 2006].

Both text-as-driver and knowledge-as-driver approaches are first-order approximations of a framework that can leverage from the powers of both knowledge and text based systems.

3 Brittleness

Brittleness is a sudden failure of a system, as opposed to a graceful degradation. The two common manifestations of brittleness are: 1) the software cannot find an answer because of gaps in its 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. The former is a bigger concern for KB systems and the latter for TB systems.

3.1 Brittleness in Knowledge

Knowledge-based systems consist of reasoning mechanisms that use an explicit *knowledge base*, a database of facts, to answer queries. The sources of brittleness in KB systems are: gaps in knowledge and inferential complexity. Commonsense knowledge was proposed as a solution to these problems [Lenat et al., 1986] in the Cyc project. Cyc's premise 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. Cyc is the largest knowledge representation effort – it consists of over 3 million assertions represented in predicate calculus. Openmind Commonsense[2], 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 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. The detailed results of this evaluation are summarized in 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 broad commonsense knowledge base is necessary for building robust programs. However, commonsense might be much vaster than imagined, and approaches to building large databases of knowledge are not enough by themselves.

3.2 Brittleness in Text

Knowledge gaps are a less serious concern for TB approaches, as a large fraction of mankind's knowledge is present in the form of text. The sources of brittlenss in text are: the limited reasoning capacity, and noise and variability of available information. However, drawing inferences and producing explanations is much harder with text. As a

[2] http://openmind.media.mit.edu/

challenge, consider the following question: How many graduate students are in Northwestern University? Further, suppose that the departmental webpages carry statistics about number of students in that department (including breakup by specializations), but the total number of graduate students is not available anywhere. One way to answer this question with this information will be to add up the students in each department. One has to make sure of two things, though: 1) no doublecounting, e.g., there might be students shared between the Computer Science and Bioinformatics departments, and 2) closed world assumption, which says that the answer assumes that we know all the departments at Northwestern University. These are straightforward to represent and reason in a logical system, however, extremely hard with text.

3.3 Heuristic Reasoning

Humans cope with the same sources of brittleness, namely, knowledge gaps and inferential complexity, with our remarkable ability to generate educated guesses, reasonable explanations and ballpark estimates when we run into situations where knowledge and/or cognitive resources are lacking. This is the insight behind the heuristic reasoning approach [Paritosh, 2006]. Heuristic methods are patterns of reasoning that yield *reasonable* answers and provide *comprehensiveness*. Reasonableness and comprehensiveness are loosely analogous to notions of soundness and completeness in formal logic. Reasonableness is what will be acceptable to a human as an answer. Similar criterion have been used in the PASCAL textual entailment challenges [Dagan et al., 2006]. A human independent measure of reasonableness is based on multiple corroborating answers: an answer is reasonable if multiple answers for the same question found using different heuristic methods are similar. Comprehensiveness captures the robustness of a set of heuristics. It is the fraction of questions of a class that the heuristics provide a reasonable answer for. Heuristic reasoning with text is a very attractive idea: there is a vast amount of broad and shallow knowledge easily available as text, which addresses the issue of knowledge gaps, and the redundancy of heuristic reasoning methods provide robustness to noise, and reliability in answers.

4 Back of the Envelope Reasoning

Back of the Envelope (BotE) reasoning is an instance of heuristic reasoning. BotE is the process of generating rough estimates to questions like, "What is the annual cost of healthcare in the US?" This type of question has been called quantification questions [Lehnert, 1986]. Quantification questions ask for a value of a quantity for some object. In many situations, 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. Additionally, if one is using sources that contain noise and can be incorrect, such reasoning serves as a sanity-checking mechanism.

We have built BotE-Solver, a system that generates back of the envelope estimates using a small set of heuristic methods, a problem solver that uses AND/OR decomposition to keep track of its progress, and the ResearchCyc knowledge base. The questions, heuristic methods and knowledge are represented in predicate calculus. Given a question, it first tries to see if the answer is already available in the knowledge base. Failing which, it uses heuristic methods to transform it into other, possibly easier, questions. This process is carried out recursively until an estimate for the original question is found. A key result from this work is that a set of seven heuristic methods achieved comprehensiveness in solving BotE problems. The details of the system and the heuristics it uses are described elsewhere [Paritosh and Forbus, 2004, 2005]. Below we present a brief description of the formalization of questions and heuristic methods.

4.1 Formalizing BotE Questions and Heuristics

We abstractly represent a question as **(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 calories are there in a Big Mac?" number of calories is the quantity and Big Mac is the object. A heuristic method transforms a question into a set of other problems **{(Q$_i$ O$_i$?V$_i$)}** such that **?V$_i$** are already known or easier to estimate. Besides transforming the problem, each heuristic method contains the answers to the questions: 1) When does it apply? and 2) How to combine **?V$_i$**s to find **?V**?

4.2 Types of BotE Heuristic Methods

There are three syntactic possibilities for a strategy based on what aspect of problem it transforms:

4.2.1 Object-based: (Q O ?V) ® {(Q O$_i$?V$_i$)}

An object-based strategy related 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 dimensions. Object-based heuristic methods include: ontology, e.g., using the information that F16 is a jet fighter craft to estimate its speed, using part-whole structure (mereology), e.g., finding the number of graduate students at Northwestern by adding up students in each department, and similarity, e.g., estimating the rent of an apartment based on a similar apartment.

4.2.2 Quantity-based: (Q O ?V) ® {(Q$_i$ O ?V$_i$)}

A quantity-based heuristic 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, i.e., **?V** and **f({?V$_i$})** have to have the same units, where `f` is the combination function. Quantity-based heuristic methods include using domain laws and rules of thumb, and using densities, e.g., estimating the national income by multiplying the per capita income by population.

4.2.3 System-based: (Q O ?V) ® {(Q_i O_i ?V_i)}

A system-based heuristic method transforms both the quantity and the object into other quantities and objects. It represents relationships between quantities of a system as a whole. System-based strategies include system laws, e.g., momentum conservation; and scale-up.

5 Textual Expressions of BotE Questions

BotE questions ask for the value of a quantitative attribute. Moriceau [2006] presents a text-based system for answering quantification question that integrates information from multiple resources, e.g., a question like “What is the average age of marriage in France?” can result in conflicting answers: varying based on the year from which the data is and whether it is for men or women. Moriceau identifies three dimensions that cause variability in answers: *time*, *place* and *restriction*. Her system identifies the cause of variance and combines the answers accordingly. It assumes that the answers are directly available in the corpus. However, consider a question like “How many people visit the Louvre each year?” If the number of people visiting the Louvre *per day* and the number of days it is open in a year are available, these two pieces of information can be combined to answer the original question. Even when there are answers directly available, BotE reasoning can help double-check the answers for plausibility.

In order to accomplish this reasoning, the question answering system has to have an understanding of the question in terms similar to that described in the previous section. First, we need to recognize that a question is a quantification question, and then extract the quantity and object in the question. A first step in many question answering systems is extraction of important entities from the question before retrieving documents [Hovy et al. 2001, Lee et al., 2001]. We present a typology of quantification questions (Figure 2)

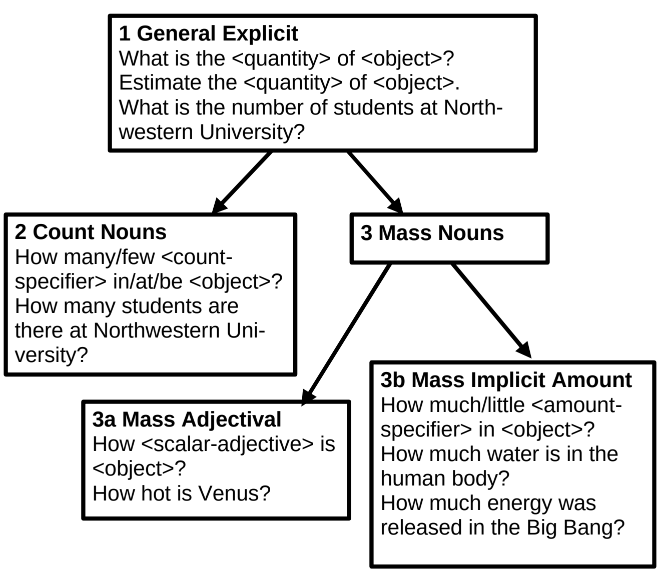

Figure 2: A typology of quantification questions.

based on an analysis of quantities and objects to guide the extraction process. The discussion below is based on English language.

5.1 Quantities

Textual expressions of quantity can be ambiguous. Consider questions like: “How big is Lake Michigan?” or “What is the cost of war in Iraq?” Big could refer to length, area, or volume, depending upon the context; and cost of war can be measured in human causalities or money spent. We will refer to a textual expression such as “big” as *quantity specifier*, and a specific attribute such as “length” as quantity.

Quantities are attributes that can be expressed numerically. Number words can be used as either nouns or adjectives. For example, in the sentence from an elementary math book, “Two and three summed are five,” the number words are nouns. This type of use of quantity is not very common outside the realm of mathematics. In normal usage, numbers are mostly used as adjectives, to describe a property, e.g., “three men.” Number words can be used to express multitudes or counts, represented by whole numbers; and magnitudes or continuous quantities, represented by real numbers. Magnitudes usually require units, e.g., “1.5 litres of milk.” Ratios, probability, and percentages are examples of dimensionless magnitudes.

As the goal of quantification questions is to find numeric answers, here we focus on explicit expressions of numeric information. Quantitative information can be expressed less directly, e.g., comparative information, e.g., A is bigger than B. Kuehne [2004] presents an analysis of quantity expressions in the domain of physical quantities. Both magnitude and multitude can be expressed using non-numeric quantifiers, e.g., “a few, many, a little, less, much, all, a lot of, enough, more, most, some, any, each, either, neither, every.” Another class of non-numeric quantifiers is gradable adjectives like “tall” and “heavy.” CARVE [Paritosh, 2004] is a computational model of this mapping between symbolic and numeric representations of quantity.

5.2 Objects

For the purposes of quantification, there are two types of objects: count nouns and mass nouns. A count noun is a noun which can be modified by a numeral and occur in both singular and plural form, as well as co-occurring with quantificational determiners like every, each, several, most, etc. A mass noun is a noun that cannot be modified by a number without specifying a unit of measurement. The count/mass distinction can be precisely defined in terms of *quantization* and *cumulativity* [Krifka, 1989]. A quantize property is such that, whenever it is true of some entity, it is not true of any proper subparts of that entity. A cumulative property is one such that if it is true both of a and b, then it is also true of the combination of a and b. Count nouns are quantized, where mass nouns are cumulative.

5.3 Questions

The general form of all quantification questions is “What is the <attribute> of <object>?” where <attribute> is a quantity. We call this the general explicit form as other textual expressions of quantification questions can be converted into to this form. It maps in a straightforward manner with our **(Q O ?V)** representation for BotE questions, where **Q** is the attribute and **O** the object. There are two classes of questions depending upon whether the object in question is a mass or a count noun.

5.3.1 Questions involving Count Nouns

Questions involving count nouns are phrased as "How many <count-specifier> in <object>?" e.g., "How many paintings are on permanent exhibit at the Louvre?" The count-specifier is the count noun that denotes the discrete unit in terms of which the enumeration is done. This question translates into "What is the *number of* <count-specifier> in <object>?" The answer to "How many" questions is a count, or a cardinality of a set or collection.

5.3.2 Questions involving Mass Nouns

Questions involving mass nouns have involve a specific dimension and assume a standard unit of measurement for that dimension. There are two forms for quantification questions involving mass nouns:

1. Mass adjectival form

The mass adjectival form is "How <gradable-adjective> is <object>?"e.g., "How tall is the Empire State building?" Gradable adjectives denote quantities that can be measured on a scale e.g., tall, rich, hot, etc. Kennedy [2003] has shown that gradable adjectives denote measure functions that map from objects to quantity values/ intervals. Questions of the mass adjectival form can be converted into the general explicit form as "What is the *quantity of* <gradable-adjective> of <object>? e.g., "What is the height of the Empire State building?" Doing this conversion requires knowing mappings between adjectives and the quantities they describe, in this example the adjective tall refers to the quantity height.

2. Mass implicit amount form

The mass implicit amount form is "How much <mass-specifier> in <object>?" e.g., "How much water is in the human body?" These questions are almost always seeking an answer that is amount or extent – size, weight, volume, etc. The mass-specifier is the mass noun whose amount in <object> is being asked for. Questions of mass implicit amount form can be translated into the general explicit form as "What is the *quantity of* <mass-specifier> in <object>?"

6 Textual Expressions of Heuristic Methods

Heuristic methods work like decomposition strategies [Katz et al., 2005a; Harabagiu and Lacatusu, 2004]. The heuristic methods we present are applicable to the entire class of BotE questions.

A requirement for applying these heuristic methods is to support some notion of variable binding. *Parameterized annotations* [Katz et al., 2005b] can be used for this purpose. Parameterized annotations combine fixed language elements with "parameters" that specify variable portions of the annotation, for example, "<*number* people live in the metropolitan area of *city*>," where the parameters are italicized. Below we describe some heuristic methods with brief discussions on how they might be implemented in text.

6.1 Ontology

Consider the question: "How fast can an F16 fly?" Even if the speed of F16 specifically is not available, the knowledge that it is a jet fighter aircraft can be used to generate an estimate. 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 **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 guard[3], 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 use the fact that Jason Kidd is a basketball player to guess his height.

Ontologies have been used by many QA systerms [Hovy et al, 2001]. However, there are some limitations to this inference: the further in the ontology hierarchy we go, the estimate is likely to less accurate. Vargas-vera and Motta [2004] present a similarity metric based on distance in the ontology that can be used to limit using this heuristic. ASIUM [Faure and Nedellec, 1998] is a system that learns an ontology from text corpora by clustering.

The second caveat in the ontology heuristic is more subtle: two basketball players might have similar height, but not necessarily two professors. This notion of what features can be inferred from a similar example is called *projectability* [Goodman, 1955]. Projectability of a feature is determined by the *centrality* of the feature. 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. It is an empirical question if centrality can be captured by statistical methods like co-occurence.

6.2 Density

Consider the question: "How many people visit the Louvre per year?" or "How many K-8 teachers are there in the US?" The density heuristic method converts a quantity into a density quantity and an extent quantity. Many "How many" questions are answered using this heuristic method. One way to answer such questions, for example the two at the beginning of this section is to enumerate them: count every instance of visitor or teacher.

Here, density is used in a general sense to mean an 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. This heuristic method exploits the fact that many numbers and statistics are more readily available as densities. If an explicit density value is not

[3] 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.

available, for example, visitors per day, then one can estimate it by looking at a typical day, or averaging the number of visitors from a set of days.

6.3 Mereology

Consider the question: "What is the number of graduate students at Northwestern University?" One way to answer this is to add up the number of students in all the departments. The mereology heuristic method transforms an object into other objects that are its parts. With regards to how quantities can be combined, they can be divided into two types: extensive and intensive. An extensive quantity is one whose value is proportional to the size of system it describes, e.g., mass, volume, etc. An intensive quantity is one whose value does not depend upon the size of the system it describes, e.g., density, temperature, etc.

If the quantity in question is an extensive parameter, then, the estimate is obtained by adding the value for the parts, as is the case in the graduate student example above. For example, the weight of a basket of fruits is the sum of weights of all the fruits and the basket. 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. This strategy requires making a closed world assumption, namely, that we know all the parts of the original object. In order for this strategy to be applicable, the parts should be non-overlapping in the quantity. If the quantity in question is an intensive parameter, then, the answer is the weighted sum, where the weights are the fraction of the whole that each part is. For example, the density of a mixture is the weighted average of the densities of the constituents.

There are many challenges in reliably implementing this strategy with text: 1) making sure we have all the parts, 2) making sure that they do not overlap, or there is no double counting, 3) finding out if the quantity is extensive or intensive. There are weaker versions of this strategy which might still be useful for purposes of double checking answers. If A is a part of B, then for any quantity, its value for A cannot be more than its value for B.

6.4 Domain Laws

Consider the basic accounting equation, *Assets = Liabilities + Shareholders' Equity*. If any two quantities are known, the third one can be computed from the formula. This strategy converts a quantity into other quantities that are related to it via laws of the domain. Domain laws include laws of physics as well as rules of thumb. For example, Newton's second law of motion, `F=m*a`, relates the force on an object to its mass and acceleration. This requires expressing the equations and formulas of interest to the QA system, and various textual expressions for each of the variables.

6.5 Similarity

The similarity heuristic transforms the object into other object(s) which are similar to it. For example, if asked for the population of Austria, a reasonable guess could be the population of Switzerland, based on the similarity of the two countries. 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. As discussed in the ontology heuristic method, this requires computing the centrality of the feature.

7 Conclusions

We believe that heuristic reasoning and integration of text and knowledge based systems are key to addressing the brittleness problem. We presented an analysis of back of the envelope reasoning and described five of the heuristics used by BotE-Solver. This paper begins to explore how heuristic reasoning could be done with text.

Acknowledgments

This research is supported by the Artificial Intelligence Program of the Computer Science Division of the Office of Naval Research. The author would also like to thank Ken Forbus for supporting these ideas and Tom Hinrichs for many discussions.

References

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.

Dagan, I., Glickman, O., and Magnini, B., 2006, The PASCAL Recognising Textual Entailment Challenge. *Lecture Notes in Computer Science*, Volume 3944, pp 177 - 190.

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*

Harabagiu, S. and Lacatusu, F., 2004, Strategies for Advanced Question Answering, Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C., 2001, Question Answering in Webclopedia, *Proceedings of the TREC-9 Conference*.

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.

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

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.

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.*

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.

Lenhart K. Schubert, 2002, "Can we derive general world knowledge from texts?", *Proceedings of the Human Language Technology Conference (HLT 2002)*, San Diego, CA, March 24-27, pp. 94-97.

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.

Miller,G. A., 1995, Wordnet: A Lexical Database for English, *Communications of the ACM*, 38(11).

Moldovan, D., Clark, C., Harabagiu, S., Maiorano, S., 2003, COGEX: A Logic Prover for Question Answering. *In Proceedings of HLT-NAACL 2003.*

Narayan, S. Harabagiu, S., 2004, Question Answering Based on Semantic Structures, *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva.

Nyberg, E., Frederking, R., Mitamura, T., Bilotti, M., Hannan, 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.*

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. 2004. Symbolizing Quantity. In *Proceedings of the 26th Cognitive Science Conference,* Chicago.

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

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

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

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*.

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.*

Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., Zhu, W. (2002). Open Mind Common Sense: Knowledge acquisition from the general public. In Robert Meersman & Zahir Tari (eds.) On the Move to Meaningful Internet Systems 2002, (pp. 1223-1237). Springer-Verlag.