Modeling the Evolution of Knowledge and Reasoning in Learning Systems

Abhishek Sharma and Kenneth D. Forbus
Qualitative Reasoning Group, Northwestern University
Evanston, IL 60208, USA
{a-sharma, forbus}@northwestern.edu

Abstract

How do reasoning systems that learn evolve over time? Characterizing the evolution of these systems is important for understanding their limitations and gaining insights into the interplay between learning and reasoning. We describe an *inverse ablation* model for studying how learning and reasoning interact: Create a small knowledge base by ablation, and incrementally re-add facts, collecting snapshots of reasoning performance of the system to measure properties of interest. Experiments with this model suggest that different concepts show different rates of growth, and that the density of facts is an important parameter for modulating the rate of learning

Introduction and Motivation

In recent years, there has been considerable interest in Learning by Reading [Barker et al 2007; Forbus et al 2007, Mulkar et al 2007] and Machine Reading [Etzioni et al 2005; Carlson et al 2010] systems. The study of these systems has mainly proceeded along the lines of measuring their efficacy in improving the amount of knowledge in the system. Learning by Reading (LbR) systems have also explored reasoning with learned knowledge, whereas Machine Reading systems typically have not, so we will focus on LbR systems here. These are evolving systems: Over time, they learn new ground facts and new predicates and collections are introduced, thereby altering the structure of their knowledge base (KB). Given the nascent state of the art, so far the learned knowledge is typically a small subset of the knowledge the system starts with. Hence the size of the KB is constant for all practical purposes, and the set of axioms it uses for reasoning will be stable and continue to perform as they did before. But what will happen to reasoning performance as the state of the art improves, and the number of facts the system has learned by reading (or using machine reading techniques) dwarfs its initial endowment?

To explore such questions, we introduce an *inverse ablation model*. The basic idea is to take the contents of a large knowledge base (here, ResearchCyc[1]) and make a simulation of the initial endowment of an LbR system by removing most of the facts. Reasoning performance is tested on this initial endowment, including the generation of learning goals. The operation of a learning component is simulated by gathering facts from the ablated portion of the KB that satisfy the learning goals, and adding those to the test KB. Performance is then tested again, new learning goals are generated, and the process continues until the system converges (which it must, because it is bounded above by the size of the original KB). This model allows us to explore a number of interesting questions, including

1. How does the growth in the number of facts affect reasoning performance? On one hand, more facts may improve the number of questions answered, since more proofs are possible. On the other hand, given that deductive reasoning is always subject to resource bounds, more facts might actually decrease the number of answers, since there are also more dead ends to explore.
2. How might the speed at which different kinds of concepts are learned vary, and what factors does that depend upon?
3. How might the number of learning goals change as the size of the knowledge base grows? Does it converge?

The inverse ablation model provides a general way of exploring the evolution of knowledge bases in learning systems. This paper describes a set of experiments that are motivated specifically by LbR systems. Under the assumptions described below, we find that (1) the size of the KB rapidly converges, (2) the growth is limited to a small set of concepts and predicates, spreading to only about 33% of the entire growth possible, (3) different concepts show different rates of growth, with the density of

[1] http://research.cyc.com

facts being an important factor in determining the rate of growth.

The rest of this paper is organized as follows: We start by summarizing related work and the conventions we assume for representation and reasoning. A detailed description of the inverse ablation model and experimental results are described next. In the final section, we summarize our main conclusions.

Related Work

A number of researchers have worked on Learning by Reading and Machine Reading systems. Learning Reader [Forbus et al 2007] used a Q/A system for evaluating what the system learned, and included *rumination*, where the system asked itself questions, using deductive and analogical reasoning to find new questions and derive new facts. Mobius [Barker et al 2007] was evaluated by comparing the facts produced by their system to a manually-generated *gold standard* set of facts. NELL [Carson et al 2010] also uses human inspection to evaluate the quality of the knowledge produced. These systems all produce formal representations. By contrast, TextRunner [Etzioni et al 2005] produces word-cluster triples. These are not formal representations that can support deductive reasoning, so they are not relevant here. A prototype system, which uses a parser and a KR&R system for deriving semantic representations of sentences for two domains, has been discussed in [Mulkar et al 2007]. Experiments related to populating the Cyc KB from the web have been described in [Matuszek et al 2005]. These systems have provided useful insights for improving our understanding of learning systems. However, measurements involving the temporal evolution of KBs and the systemic properties of rapidly changing learning systems have not been the focus of these endeavors. In addition to LbR research, our work is inspired by the literature on the evolution of the World Wide Web [Ntoulas et al 2004], graphs [Leskovec et al 2007] and social networks [Kossinets & Watts 2006].

Representation and Reasoning

We use conventions from Cyc [Matuszek *et al* 2006] in this paper since that is the major source of knowledge base contents used in our experiments[2]. We summarize the key conventions here. Cyc represents concepts as *collections*. Each collection is a kind or type of thing whose instances share a certain property, attribute, or feature. For example, Cat is the collection of all and only cats. Collections are arranged hierarchically by the `genls` relation. `(genls <sub> <super>)` means that anything that is an instance of <*sub*> is also an instance of <*super*>. For example, `(genls Dog Mammal)` holds. Moreover, `(isa <thing> <collection>)` means that <*thing*> is an instance of collection <*collection*>. Predicates are also arranged in hierarchies. In Cyc terminology, `(genlPreds <s> <g>)` means that <*g*> is a generalization of <*s*>. For example, `(genlPreds touches near)` means that touching something implies being near to it. The set of `genlPreds` statements, like the `genls` statements, forms a lattice. In Cyc terminology, `(argIsa <relation> <n> <col>)` means that to be semantically well-formed, anything given as the <*n*>th argument to <*relation*> must be an instance of <*col*>. That is, (<*relation*>……<*arg-n*> …) is semantically well-formed only if `(isa <arg-n> <col>)` holds. For example, `(argIsa mother 1 Animal)` holds.

Learning by Reading systems typically use a Q/A system to examine what the system has learned. For example, Learning Reader used a parameterized question template scheme [Cohen *et al*, 1998] to ask ten types of questions. The templates were: (1) Who was the actor of <*Event*>?, (2) Where did <*Event*> occur?, (3) Where might <*Person*> be?, (4) What are the goals of <*Person*>?, (5) What are the consequences of <*Event*>?, (6) When did <*Event*> occur?, (7) Who was affected by the <*Event*>?, (8) Who is acquainted with (or knows) <*Person*>?, (9) Why did <*Event*> occur?, and (10) Where is <*GeographicalRegion*>? In each template, the parameter (e.g., <*Person*>) indicates the kind of thing for which the question makes sense (specifically, a collection in the Cyc ontology). We use these questions in our experiments below, to provide realistic test of reasoning.

When answering a parameterized question, each template expands into a set of formal queries, all of which are attempted in order to answer the original question. Our FIRE reasoning system uses backchaining over Horn clauses with an LTMS [Forbus & de Kleer 93]. We limit inference to Horn clauses for tractability. We use network-based optimization techniques for automatically selecting an efficient set of axioms. Inference is limited to depth 5 for all queries, with a timeout of 90 seconds per query.

An Inverse Ablation Model

Deductive reasoning is one of the principle reasons for accumulating large knowledge bases. In large knowledge-based systems, inference engines generate and examine thousands of potential proof paths for answering target queries. Understanding how deductive inference performance changes as KBs grow is the fundamental motivation for the inverse ablation model. Since large-scale learning systems are in their infancy, instrumenting a learning system that is operating over months is still not possible. Hence we start by ablating a large KB and measure reasoning performance as we add knowledge back in. The parameters of an inverse ablation model include (1) what is the initial endowment? (2) what reasoning

[2] We use knowledge extracted from the ResearchCyc knowledge base with our own reasoning system, instead of using Cycorp's reasoning system.

methods are used?, (3) How are queries generated, and (4) what is the strategy used to grow the knowledge base? We discuss each of these decisions in turn.

Initial endowment: Since we are using ResearchCyc contents, the initial endowment consists of the basic ontology definitions (the `BaseKB` and `UniversalVocabularyMt` microtheories) plus 5,180 facts chosen at random. This leaves 491,091 facts that could be added on subsequent iterations to simulate learning. We refer to this collection of facts as the *fact repository*, to distinguish it from the KB used in reasoning in a learning iteration. One interesting measure is how much of the fact

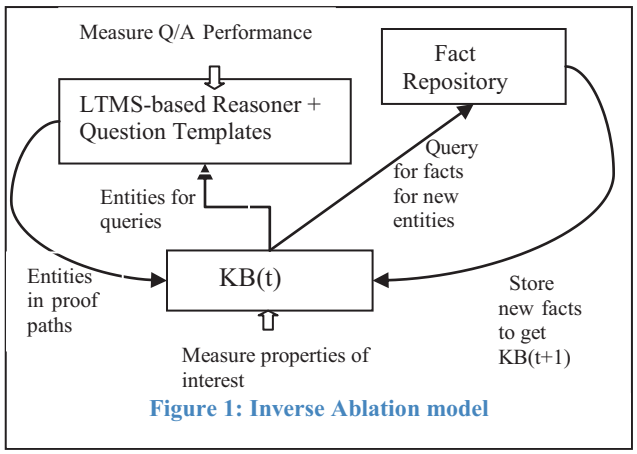

Figure 1: Inverse Ablation model

repository ends up being added back when the system converges: Facts that remain in the repository at that point have no perceived relevance to the questions that are driving learning.

Reasoning method: CSP solvers [van Dogngen et al 2009] are arguably the most efficient solvers available today, but are limited to propositional reasoning, making them inappropriate for open domains and large-scale worlds where propositionalization would lead to an exponential explosion in the number of axioms. By contrast, Cyc systems such as ResearchCyc and OpenCyc include broadly capable reasoning engines that handle a wide variety of higher-order constructs and modals, making them very flexible, at the cost of efficiency. The Horn-clause backchaining strategy outlined above represents a compromise between these extremes that we have found useful in a variety of systems[3]. Since that is the method we use in our own LbR research, that is what we use in this model.

Query Generation: We automatically generate a set of queries at each iteration by asking every question for every entity that satisfies the collections associated with each type of parameterized question. Thus the types of entities, given the set of parameterized questions, are `Event`, `Person`, and `GeographicalRegion`. Note that as the KB grows, so too can the number of queries generated, since new entities of these types can be added. This allows us to measure how costly different strategies for generating learning goals might be.

Algorithm

1. Set t← 0.
2. Initialize KB(t) by choosing facts randomly from the repository.
3. Repeat step 4 until the process converges.
4. loop
   a. Set Q ← Generate all questions for the question templates mentioned on page 2.
   b. Ask the set of questions Q and measure Q/A performance.
   c. E ← the set of entities in intermediate queries generated during the reasoning process.
   d. Let *Facts* ← New facts about the elements of E in the Fact Repository.
   e. KB(t+1) ← KB(t) + *Facts*
   f. Record the properties of interest for KB(t+1)
   g. If ΔKB → 0 then exit loop, else t ← t+1and go to step 4(a).

Figure 2: Inverse Ablation Model

Growth Strategy: The method for growing the KB by adding back in facts should reflect assumptions made about the way the system generates learning goals. At each iteration, we use reasoning failures to generate learning goals, which are then used to gather facts from the fact repository. Specifically, the proof trees for failed queries are examined to find nodes representing queries involving specific entities. Finding out more about these entities become the learning goals for that iteration. For example, a query like `(acquaintedWith BillClinton ?x)` leads to an intermediate query like `(mother ChelseaClinton ?x)`. Hence learning about `ChelseaClinton` would become one of the learning goals for that iteration.

We model the effect of learning by gathering all of the facts which mention the entities in learning goals from the fact repository. This is tantamount to assuming a large amount of learning effort in every cycle, essentially mining out everything that is going to become known about an entity the first time that it becomes a target for learning. While optimistic, pursuing any other strategy would require making more assumptions, thereby making them harder to justify. This gives us an extreme point, at least.

Figure 1 shows a schematic diagram of how the inverse ablation model works, and Figure 2 describes the

[3] It corresponds closely to Prolog, albeit without cut and without clause ordering.

Time (t)	No. of Ground Facts in KB	No. of Questions	No. of Queries	No. of Answers	%	Time (min.)	Time per query (min)	% increase in no. of ground facts	% increase In Q/A	% increase in time/query
0	5180	4023	7428	3	0.07	28	0.003	-	-	-
1	66171	4794	9663	1426	29.70	75	0.007	1177	423	133
2	143922	12943	33198	4853	37.49	566	0.017	2678	534	466
3	159298	14114	36615	5984	42.39	702	0.019	2975	604	533
4	165050	14584	37965	6480	44.43	735	0.019	3086	633	533
5	165992	14645	38148	6548	44.71	759	0.019	3104	637	533

Figure 5: Q/A Performance

experimental procedure used, in algorithmic form. Step 4(a) extracts all entities needed for instantiating the question templates described in Section 3. Step 4(b) uses backchaining to attempt to answer these queries, and records the Q/A performance, including the partial proof paths and intermediate queries. The set *E* in step 4(c) refers to the entities in these queries. These entities are sent to the fact repository and all facts involving them are collected. Step 4(e) adds these facts to KB(t) to get KB(t+1).

Experimental Results

This section discusses the results of running the procedure in Figure 2. Figure 3 shows the change in number of ground facts. The number of facts increases rapidly from 5,180 at t=0 to 143,922 facts at t=2. The curve asymptotes to about 166,000 facts at t=5. It is also useful to compare the extent of this growth with respect to the contents of fact repository. The coverage increases from 1% of the repository at t=0 to 33% at t=5. The high rate of growth shows that the domain is densely connected and the average distance between two nodes is pretty small. On the other hand, given these questions, about 67% of the repository is beyond our reach. Next, we turn to the rate of introduction of new predicates and concepts (see Figure 4). At t=0, 53% of the predicates had least one ground fact associated with them. After five learning iterations, 65% predicates had at least one ground fact. Similarly, the proportion of concepts with at least one instance increased from 53% to 62%. This shows that the new facts are being drawn from a small set of predicates and concepts.

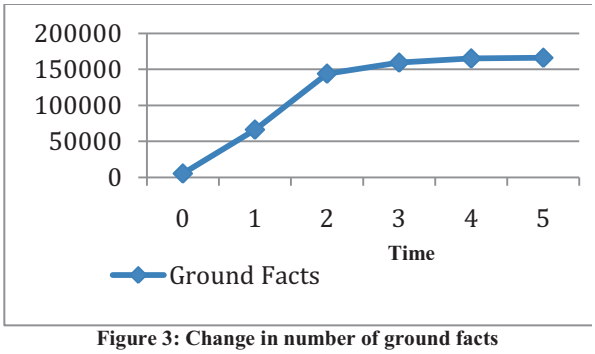

Figure 3: Change in number of ground facts

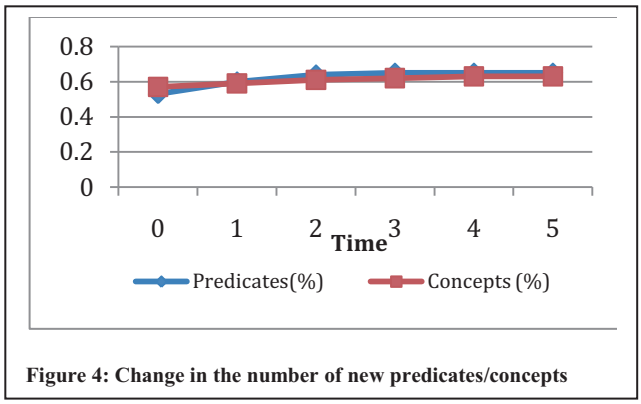

Figure 4: Change in the number of new predicates/concepts

In Figure 5, the dynamics of Q/A performance is shown. The last three columns of the table show improvement with respect to the KB at t=0. The proportion of questions answered improves significantly with the size of the KB. While the size of KB increased by 3,104% in five iterations, the proportion of questions answered increased by 637%. The time needed per query increased by 533% during this period. These results suggest that time-constrained deductive reasoning systems would need new methods to select the best set of axioms due to increasing resource requirements and changing distribution of facts and collections.

It is also interesting to compare the rate of growth of different regions of the KB and check if some of them display unusual patterns. Recall that the question types discussed involve three kinds of concepts: `Person`, `Event` and `GeographicalRegion`. We measured the rate of growth of instances of these concepts and found that they vary greatly. In Figure 6, we see that the KB had 1.4% of all instances of `Person` at t=0. This grew to 2% after five iterations. During the same period, the proportion of `GeographicalRegion` increased from 7.9% to 58%. The proportion of instances of `Event` grew from 26% to 33%. It shows that the rate of growth of `GeographicalRegion` is

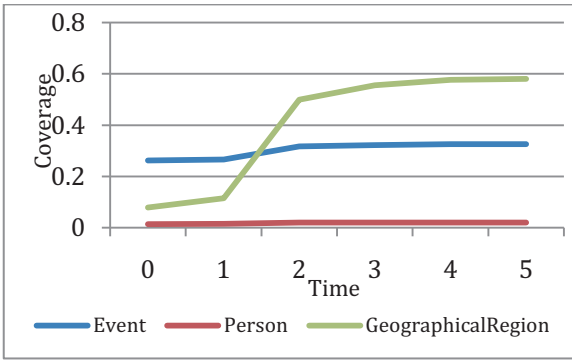

Figure 6: Growth of different concepts

pretty high, whereas this model has not made significant progress in accumulating knowledge about instances of `Person`. One important reason for this difference is the density of facts for these concepts. In Figure 7, we show the cumulative distribution of number of facts per entity for these concepts. The x-axis shows the number of facts per entity for instances of each of these three concepts.

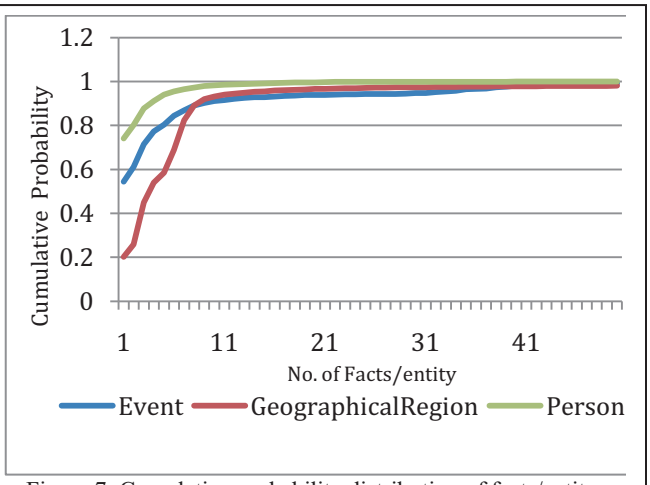

Figure 7: Cumulative probability distribution of facts/entity

The y-axis shows the cumulative probability. In Figure 8, we compare the mean and median facts/entity and growth in coverage of three concepts. When the mean facts per entity increases from 2.14 to 5.58, the growth rate changes from 0.5% to 6.2%. This shows that the density and the rate of growth show a nonlinear relationship and it can be used to modulate the rate of learning. In Figure 9 and 10, we study the change in distribution of number of instances for the specializations of `GeographicalRegion` and `Event`. For understanding the extent of this evolution, we have also included the distribution of instances for the fact repository. The x-axis shows the number of instances in the collection. The y-axis shows the cumulative probability. In Figure 9, we see that the distribution

Concept	Mean facts per entity	Median facts per entity	Growth in coverage
Person	2.14	1	0.5%
Event	5.58	2	6.2%
GeographicalRegion	11.29	5	50.1%

Figure 8: Comparison of mean and median facts/entity and growth

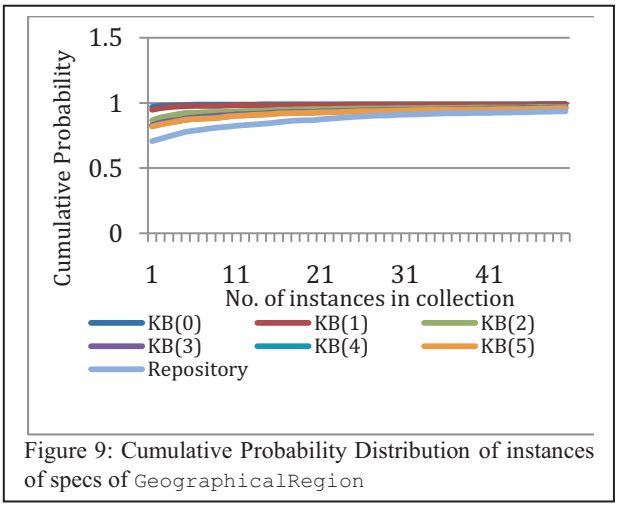

Figure 9: Cumulative Probability Distribution of instances of specs of `GeographicalRegion`

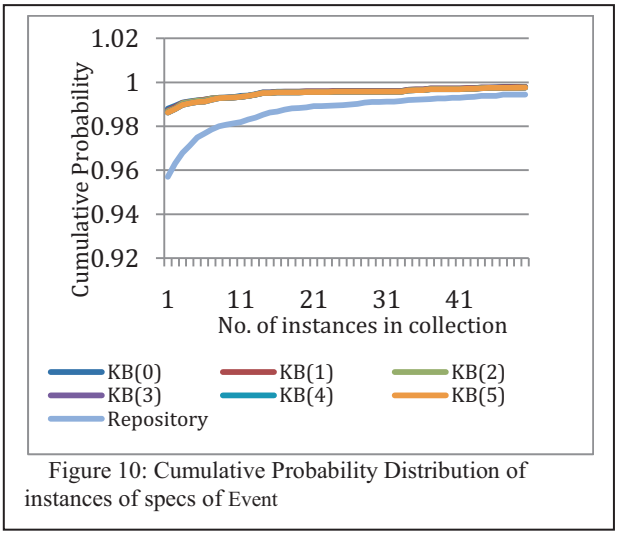

Figure 10: Cumulative Probability Distribution of instances of specs of Event

of KB(5) is significantly different from the distribution of KB(0). Moreover, the distribution of KB(t) (where $0 \leq t \leq 5$) has steadily moved towards the distribution of the fact repository. On the other hand, Figure 10 shows that the distribution of instances in KB(t) (where $0 \leq t \leq 5$) has not changed much and is very different from the distribution in the fact repository.

Conclusion

There has been growing interest in creating large-scale learning systems, such as Learning by Reading systems. However, there has been relatively little work in studying the properties of reasoning systems which grow significantly over time. We have proposed an inverse ablation model for studying how reasoning performance changes with KB growth, as might be caused by learning. The method proposed here is very general and could be used with any large KB or KR&R system. We have studied performance aspects of the evolving KB that are of particular interest from the perspective of learning systems. The model proposed here increased the size of the KB from 1% to 33% of the repository in five iterations. As the number of facts, predicates and collections increase, the size of search space and dynamics of reasoning would change as well. This implies that learning algorithms and inference engines should use distribution sensitive algorithms which would adapt well to a changing KB. Growth is compartmentalized but spreads to a significant fraction of the fact repository. Growth is focused, as indicated by the new facts being about a small number of predicates and concepts. Different concepts show different rates of growth, which can be explained by their densities. Our results show that the rate of growth in high density regions is very high. The total number of queries increased from 7,428 to 38,148; and the total time needed to answer the queries increased from 28 minutes to 759 minutes. Such an increase may affect the usability of the system. Therefore, similar systems may need to design appropriate parameters for controlling growth in high density regions. On the other hand, increasing the knowledge about low density regions is a challenge. In a sparsely connected domain, systems like ours may need to find ways to hop from one island to another using other learning methods.

Acknowledgements

This research was funded by the Intelligent Systems program of the Office of Naval Research.

References:

K Barker, B Agashe, S Y Chaw, J Fan, N. Friedland, M Glass, J. Hobbs, E. Hovy, D. Israel, D Kim, R Mulkar-Mehta, S Patwardhan, B Porter, D Tecuci, P Z. Yeh: Learning by Reading: A Prototype System, Performance Baseline and Lessons Learned. *Proc. of AAAI* 2007: 280-286

P. Cohen, Schrag, R., Jones, E., Pease, A., Lin, A., Starr, B., Gunning, D., and Burke, M. 1998. The DARPA High-Performance Knowledge Bases Project. AI Magazine, 19(4), Winter, 1998, 25-49

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. and Mitchell, T. 2010. Toward an architecture for Never-Ending Language Learning. Proceedings of AAAI-10.

van Dongen, M., Lecourte, C. and Rousell (Eds.) *Proceedings of the 3rd International CSP Solver Competition*.

O Etzioni, M J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, Alexander Yates: Unsupervised named-entity extraction from the Web: An experimental study. *Artif. Intell.* 165(1): 91-134 (2005)

K. D. Forbus and J. de Kleer. *Building Problem Solvers*. MIT Press, 1993

K D. Forbus, C Riesbeck, L Birnbaum, K Livingston, A Sharma, L Ureel II: Integrating Natural Language, Knowledge Representation and Reasoning, and Analogical Processing to Learn by Reading. *Proc . of AAAI* 2007: 1542-1547

G. Kossinets and D. Watts. Empirical Analysis of an Evolving Social Network. *Science,* 311, 88, 2006

J. Leskovec, J. Kleinberg and C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters, *ACM Trans. on Knowledge Discovery from Data,* Vol 1, No. 1, 2007.

C. Matuszek, J. Cabral, M. Witbrock, J. DeOliveira, An Introduction to the Syntax and Content of Cyc, *AAAI Spring Symp. on Formalizing and Compiling Background Knowledge and Its Applications to KR and QA,* CA, March 2006.

C. Matuszek, M. Witbrock, R. Kahlert, J. Cabral, D. Schneider, P Shah, D. B. Lenat, Searching for Common Sense: Populating Cyc from the Web, *Proc. of AAAI*, 2005.

R. Mulkar, J. Hobbs, E. Hovy, H. Chalupsky and C. Lin Learning by Reading: Two Experiments. Third Intl. Workshop on Knowledge and Reasoning for Ans. Questions, 2007

A. Ntoulas, J Cho and C. Olston. What's New on the Web? The Evolution of the Web from a Search Engine Perspective. *Proc. of WWW*, 2004.