Modeling the Evolution of Knowledge in Learning Systems

Abhishek Sharma[1] Kenneth D. Forbus[2]

[1]Cycorp, Inc. 7718 Wood Hollow Drive, Suite 250, Austin, TX 78731
[2]Northwestern University, 2133 Sheridan Road, Evanston, IL 60208
e-mail: abhishek@cyc.com, forbus@northwestern.edu

Abstract

How do reasoning systems that learn evolve over time? What are the properties of different learning strategies? 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 large knowledge-based systems evolve: Create a small knowledge base by ablating a large KB, and simulate learning by incrementally re-adding facts, using different strategies to simulate types of learners. For each iteration, reasoning properties (including number of questions answered and run time) are collected, to explore how learning strategies and reasoning interact. We describe several experiments with the inverse ablation model, examining how two different learning strategies perform. Our results suggest that different concepts show different rates of growth, and that the density and distribution of facts that can be learned are important parameters 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 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 small compared to the knowledge base the system starts with. Hence the learning trajectory and final state of the system is known for all practical purposes. But what will be the learning trajectory 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? (2) How might the speed at which different kinds of concepts are learned vary, and what factors does that depend upon? (3) Is learning focused, or are we learning facts about a wide range of predicates and concepts? (4) What are the properties of different learning strategies? (5) How does the distribution of facts that can be acquired affect the learning trajectory?

The inverse ablation model provides a general way to explore the evolution of knowledge bases in learning systems. This paper describes a set of experiments that are motivated 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 facts being an important determining factor, and (4) Different learning strategies have significant differences in their performance, and the distribution of facts that can be learned also plays an important role.

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

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

Abhishek Sharma and Kenneth D. Forbus. Modeling the Evolution of Knowledge in Learning Systems. *Proceedings of AAAI 2012, Twenty Sixth Conference On Artificial Intelligence, Toronto, Canada*

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*. 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. In 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 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]. These systems focus on the changing structure of graphs and networks. We believe that knowledge-based systems should be cognizant of how learning algorithms affect the structure of the KB. This knowledge will help them to choose those learning trajectories which would lead to more efficient KB structure.

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*>.

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). The queries that each template expand to all contain exactly one open variable, whose binding is found

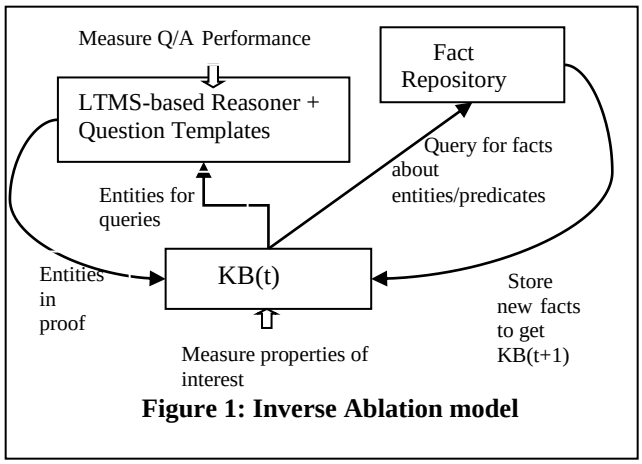

Figure 1: Inverse Ablation model

via inference in order to answer the question (e.g. for Q1, the answer variable is bound to a person). If BillGates and BillClinton are the instances of Person in KB then question type 3 would expand to {(objectFoundInLocation BillGates ?x),(objectFoundInLocation BillClinton ?x)}. We use these questions in our experiments below, to provide a realistic and relevant test of reasoning.

An Inverse Ablation Model

Deductive reasoning is a primary reason for accumulating large knowledge bases[3]. 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

[2] We use a subset of ResearchCyc knowledge base with our FIRE reasoning system [Forbus & de Kleer 1993].

[3] We are referring to the goal of building machines capable of doing various tasks which need common sense. It is generally accepted that such systems must be able to reason deductively with large amounts of data.

Abhishek Sharma and Kenneth D. Forbus. Modeling the Evolution of Knowledge in Learning Systems. *Proceedings of AAAI 2012, Twenty Sixth Conference On Artificial Intelligence*, Toronto, Canada

measure reasoning performance as we add knowledge back in. Figure 1 shows a schematic diagram of how the inverse ablation model works. The parameters of an inverse ablation model include (1) What is the initial endowment? (2) What reasoning methods are used?, (3) How are queries generated?, (4) What is the distribution of facts in the external knowledge source?, and (5) What is the strategy used to grow the knowledge base? We discuss each decision in turn.

Initial endowment: Since we are using ResearchCyc contents, the initial endowment consists of the basic ontology definitions (the BaseKB and UniversalVocabularyMt microtheories[4]) plus about 5,000 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 during a learning iteration. One interesting measure is how much of the fact 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 are arguably the most efficient solvers available today, but are largely 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 include broadly capable reasoners that handle a wide variety of higher-order constructs and modals, making them very flexible, at the cost of efficiency. The reasoning system we use here is FIRE because it was used in the Learning Reader system [Forbus et al 2007]. FIRE performs backchaining over Horn clauses, similar to Prolog but without clause ordering or cut, and uses an LTMS [Forbus & de Kleer 93] for caching answers. Following Learning Reader, inference is limited to depth 5 for all queries, with a timeout of 90 seconds per query[5]. Each parameterized question template is expanded into a set of formal queries, all of which are attempted in order to answer the original question.

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.

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. Moreover, it is also interesting to study the properties of different learning strategies. Below we compare the performance of two strategies:

(1) *Entity-based Learning Strategy*: 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 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[6].

(2) *Predicate-based Learning Strategy*: While using this strategy, the reasoner chooses a new predicate *pred* in every learning iteration, which would lead to maximum improvement in Q/A performance. All facts matching the pattern (pred ?x ?y) are sought from the fact repository. The algorithm used for assessing the utility of learning about predicate *m* is shown in Figure 2. Here NumberOfFacts(p) represents our estimate of the number of facts which can be inferred about predicate p. In step 1 of the algorithm, we

Algorithm: ***ReturnEstimateOfPerformance***
Input: Predicate *m*

1. For all predicates p in the KB do:
   a. *NumberOfFacts*(p)←*KBFacts*(p)
2. *NumberOfFacts*(m)←*KBFacts*(m)+N
3. *OrderedList* ← Perform a topological sort of the directed search space represented by the axioms. Break cycles arbitrarily.
4. For each p in *OrderedList*
   $NumberOfFacts(p) \leftarrow \sum_Q NumberOfFacts(q) + \sum_R \prod_S \alpha * NumberOfFacts(s)$
5. Return $\sum_{RootNodes} NumberOfFacts(r)$

Figure 2: Algorithm for assessing the utility of learning

[4] These microtheories were chosen because they contain the definitions of predicates and collections. It would not be possible to generate questions without the generalization hierarchy. However, including the hierarchy precludes the possibility of studying the evolution of its structure.

[5] Due to the large size of search space, such parameters are used by most inference engines.

[6] Obviously mining everything known about an entity is only feasible when the external knowledge source is small. In other cases, computational constraints will impose an upper limit on the number of ground facts acquired by the system.

initialize it by KBFacts(p), which represents the number of ground facts in the KB about predicate *p*. In step 2, we assume that we can get N facts[7] (In this work, N was set to 1,000.) from the fact repository about predicate *m*. A topological sort of the directed search space is performed in step 3 and inference is propagated bottom-up in step 4. The first-term on the RHS of step 4(a) sums evidence from the OR nodes which are the children of p. The second term adds evidence from those AND nodes which are the children of p. The constant α represents the probability of unification and was set to 10^{-3}. Step 5 returns the number of answers for the root nodes (i.e., the target queries). The process is repeated for all predicates and the predicate which return the maximum value is sent to the fact repository as the learning query. We chose this strategy for our experiments because predicates play an important role in any KR&R system. Secondly, given a search space represented by axioms, it is natural to represent the learning requests in the form of nodes, which are automatically mapped to predicates.

Algorithm

1. **Input:** Growth Strategy (Entity-based or Predicate-based)Set t← 0.
2. Initialize KB(t) by choosing facts randomly from the repository.
3. Repeat steps 4 to 9 until the process converges (i.e., ΔKB → 0)
4. Set *Q* ← Generate all questions for the question templates mentioned on page 2.
5. Ask the set of questions *Q* and measure Q/A performance.
6. If the Growth Strategy is *Entity-based* then:
   a. *E* ← the set of entities in intermediate queries generated during the reasoning process.
   b. Let *Facts* ← New facts about the elements of *E* in the Fact Repository.
7. Else if growth strategy is *Predicate-based*
   a. Choose a predicate *p* from the search space which would lead to the maximum gain in Q/A performance.
   b. Let *Facts* ← New facts which match the pattern `(p ?x ?y)` from the Fact Repository.
8. KB(t+1) ← KB(t) + *Facts*
9. Record the properties of interest for KB(t+1)

Figure 3: Inverse Ablation Model

Figure 3 describes the experimental procedure used, in algorithmic form. Step 6 describes the entity-based learning strategy, while Step 7 describes the predicate-based strategy. Q/A performance (i.e., proportion of questions answered) is recorded in step 5, whereas systemic properties of the KB (shown in Figures 5, 6 and 8) are measured in step 9.

Distribution of Facts in the External Knowledge Source: It is useful to view the knowledge base as a giant graph where entities are connected via different kinds of relations. Given this representation, some parts of the graph are more densely connected than others[8]. Note that the degree of a given node in this graph corresponds to the number of facts associated with it[9]. Therefore, the trajectory of learning in the entity-based strategy is determined by the number of facts acquired from entities. Similarly, the number of facts acquired from predicates determines the trajectory for the predicate-based strategy. In Figure 4, we show the proportion of predicates and entities which have their number of facts in a given range. For example, the graph shows that the probability of a randomly chosen entity having 0-5 facts is 0.13. On the other hand, the probability that a randomly chosen predicate would have 0-5 facts is more than 0.5. The graph should be seen as a descriptive summary of the density of the fact repository. The last bin in Figure 4 shows the probability mass not included in other bins (i.e. Pr(x> 50)). The difference shown in Figure 4 causes the differences in the properties of learning strategies shown in next section.

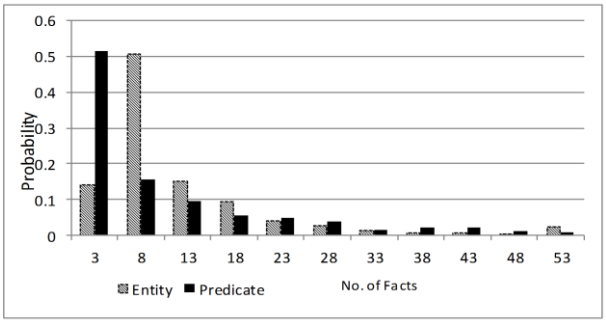

Figure 4: Distribution of facts in external knowledge source

Experimental Results

The experiments were done for three starting points for KB(0). Since the results for these experiments were similar, we report average of these results in Figures 5 to 10. Figure 5 shows the change in number of ground facts. For the entity-based model, the number of facts increases rapidly from 4,968 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

[7] The value of N used here is an estimate. However, building perfect models of an unseen repository is an interesting research problem.

[8] Generally density is measured over a volume. However, in graph theory, a *dense graph* is a graph in which the number of edges is close to the maximal number of edges.

[9] Assume that the KB contains following two facts: {(wife HillaryClinton BillClinton), (daughter ChelseaClinton HillaryClinton)}, then HillaryClinton has 2 facts associated with her. BillClinton and ChelseaClinton have just one fact associated with them. Therefore the degree of the HillaryClinton node would be two. The degree of other two nodes is one.

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.

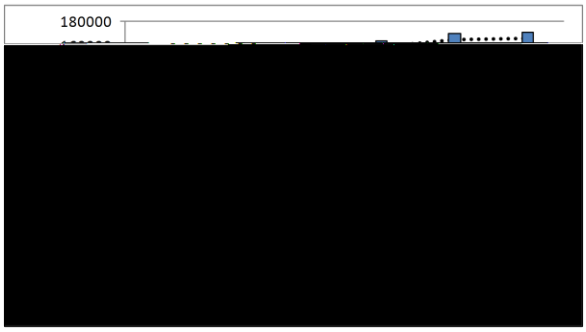

Figure 5: Change in Number of Ground Facts (Average of three experiments).The results are statistically significant ($p<0.05$).

On the other hand, given these questions, about 67% of the repository is beyond our reach. The number of facts asymptotes at 5% of the fact repository for the predicate-based model. Next, we turn to the rate of introduction of new predicates and concepts (see Figure 6). In this case, both learning strategies showed similar performance. At t=0, about 55% of the predicates had least one ground fact associated with them. After five learning iterations, 65% predicates had at least one ground fact[10].

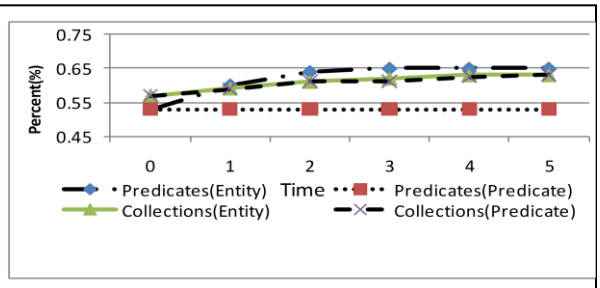

Figure 6: Change in the number of new predicates/concepts (Average of three experiments.)

Similarly, the proportion of concepts with at least one instance increased from 53% to 62%. This shows that the learning is focused and new facts are being drawn from a small set of predicates and concepts. It also points towards homophily[11] in the ground facts because many different concepts are out of our reach. In Figure 7, the dynamics of Q/A performance is shown. The proportion of questions answered improves significantly with the size of the KB. For the entity-based model, the size of KB increased by 3,104% in five iterations, but the proportion of questions answered increased by only 637%. The time needed per query increased by 533% during this period.

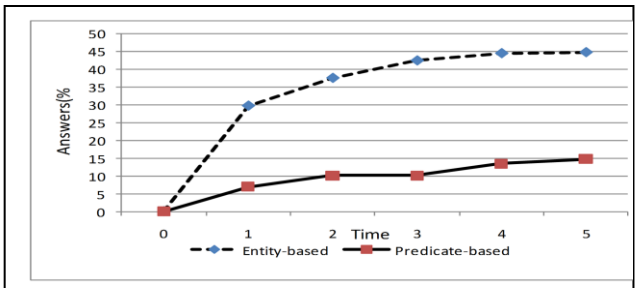

Figure 7: Q/A performance (Average of three experiments). The results are statistically significant. ($p<0.05$)

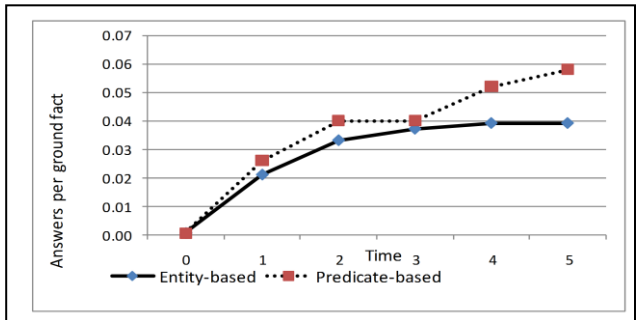

Figure 8: Number of answers per unit ground fact (Average of three experiments).

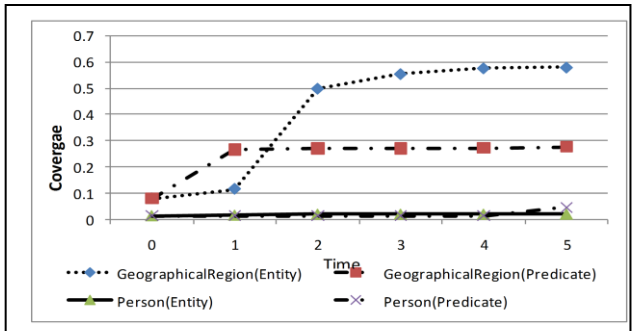

Figure 9: Growth of different concepts (Average of three experiments). The results are statistically significant. ($p<0.05$)

These results suggest that time-constrained deductive reasoning systems will need new methods to select the best set of axioms due to increasing resource requirements and changing distribution of facts and collections. The entity-

[10] Let us assume that the KB has following three facts: (doneBy Speaking-100 BillClinton), (doneBy Killing-209 Person-198) and, (isa Canada Country). Then the number of ground facts associated with predicates doneBy and isa are 2 and 1 respectively. Similarly, the collection Country has just one instance.

[11] In this context, homophily refers to the phenomenon of entities linking to similar entities via relations.

based model performs better than the predicate-based model as far as net Q/A coverage is concerned. On the other hand, Figure 8 shows that the predicate-based model uses fewer facts to derive its answers than the entity-based model. The number of question answered per ground fact also improves with time for both strategies[12].

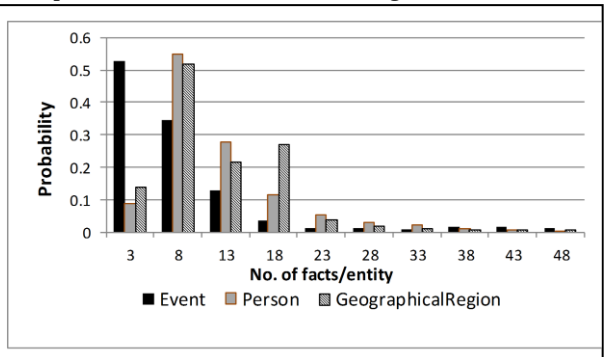

Figure 10: Probability distribution of facts/entity for concepts

	Entity-based	**Predicate-based**
No. of Facts	33% of maximum	5% of maximum
Focused Learning	Yes	Yes
Q/A (%)	Better	Worse
Utilization of ground facts for deriving answers	Worse	Better
Distribution of Learning	Skewed	Uniform

Table 1: Summary of key differences

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[13]. The predicate-based model did not show any significant difference in growth patterns in different regions of the KB (see Figure 9). However, the rates of growth of instances of these concepts vary greatly for the entity-based model. In the figure below, 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% (not shown in Figure). It shows that the rate of growth of GeographicalRegion is 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 10, we show the 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. The mean of facts per entity for Person, Event and GeographicalRegion are 2.14, 5.58 and 11.29 respectively. The medians of facts for these concepts are 1, 2 and 5 respectively. The net growth in coverage for these concepts was 0.5%, 6.2% and 50.1% respectively. 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 Table 1, we summarize key differences between entity-based and predicate-based strategies.

Conclusion and Discussion

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. The results show significant differences between the performances of two learning strategies that are of particular interest from the perspective of learning systems. The points of convergence of learning strategies pose interesting questions for design of learning strategies. In particular, what kind of strategies would help us to acquire facts from all parts of the KB? Secondly, how should we change these strategies to ensure that we can get close to 100% facts from the repository?

The interplay of learning and reasoning is visible in the properties of entity-based learning strategy. Seeking facts about entities in reasoning paths leads to higher growth in denser regions. This positive feedback leads us to believe that such learning systems would enhance their knowledge of those domains about which the KB already contains some minimum threshold of knowledge. For other domains, the level of knowledge would stagnate. In a sparsely connected domain, learning systems may need to find ways to hop from one island to another using other learning methods. How can we use this analysis up front? If we already know the distribution of knowledge in the external source, then we might control the rate of learning by designing appropriate parameters. The task is more difficult if the distribution is unknown. Since we cannot make the assumption that we have a fair model of the

[12] In this case, the differences between the performances of two strategies are less significant and pronounced. We intend to study this issue in more detail in future work. However, it is interesting to note that the number of answers per ground fact increases with the size of KB in both cases.

[13] These concepts were chosen because they are general enough to include thousands of other concepts via the generalization hierarchy.

complete external source, we will have to continuously revise our approximation of distribution of knowledge. The inference engine should use it for better allocation of resources.

The differences discussed above are due to the different probability distributions of entities and predicates in the external knowledge source[14] (Figure 4). We observed that one of the models 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 in order to 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. Given these results, we believe that the entity-based strategy should be preferred over the predicate-based strategy. However, understanding how best to combine these strategies and use the distribution of facts to achieve a balanced and efficient learning trajectory is an interesting open question. How would one decide on a policy for KB growth? We believe that minimizing time requirements for reasoning is critical for large knowledge-based systems. Therefore, we should choose a strategy which satisfies Q/A performance thresholds by acquiring facts about minimum number of predicates. This would help us to minimize time requirements because we would be able to choose a small number of axioms to reason with them. Applying these learning strategies in a new learning by reading system is something we plan to do in future work.

Acknowledgements

The contents of this paper benefitted from discussions with Tom Hinrichs. This work was supported by the Intelligent Systems Program of the Office of Naval Research.

References

K Barker, B Agashe, S Y Chaw, et al. Learning by Reading: A Prototype System, Performance Baseline and Lessons Learned. *Proc. of AAAI* 2007: 280-286

P. Cohen, Schrag, R., Jones, E., Pease, et al. 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, 2010.

O Etzioni, M J. Cafarella, Doug Downey, et al. 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.

[14] Other learning systems can expect similar results if they use an external fact repository whose topological and distributional properties are similar to the corresponding properties of our fact repository. Generalizing these results to other cases is an interesting research problem.