Strategies for Lifelong Knowledge Extraction from the Web

Michele Banko and Oren Etzioni
Turing Center
University of Washington
Computer Science and Engineering
Box 352350
Seattle, WA 98195, USA
banko@cs.washington.edu, etzioni@cs.washington.edu

ABSTRACT
The increasing availability of electronic text has made it possible to acquire information using a variety of techniques that leverage the expertise of both humans and machines. In particular, the field of Information Extraction (IE), in which knowledge is extracted automatically from text, has shown promise for large-scale knowledge acquisition.

While IE systems can uncover assertions about individual entities with an increasing level of sophistication, text understanding – the formation of a coherent theory from a textual corpus – involves representation and learning abilities not currently achievable by today's IE systems. Compared to individual relational assertions outputted by IE systems, a *theory* includes coherent knowledge of abstract concepts and the relationships among them.

We believe that the ability to fully discover the richness of knowledge present within large, unstructured and heterogeneous corpora will require a *lifelong learning* process in which earlier learned knowledge is used to guide subsequent learning. This paper introduces ALICE, a lifelong learning agent whose goal is to automatically discover a collection of concepts, facts and generalizations that describe a particular topic of interest directly from a large volume of Web text. Building upon recent advances in unsupervised information extraction, we demonstrate that ALICE can iteratively discover new concepts and compose general domain knowledge with a precision of 78%.

K-CAP'07, October 28–31, 2007, Whistler, British Columbia, Canada.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge Acquisition*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Control theory*

General Terms
Algorithms

1. INTRODUCTION
The accumulation of knowledge takes on a variety of forms, ranging from the provision of information by domain experts and volunteer contributors, to techniques that extract information from structured data, to systems that seek to understand natural language. Driven by the increasing availability of online information and recent advances in the fields of machine learning and natural language processing, the AI community is increasingly optimistic that knowledge capture from text is within reach [4, 9]. Unsupervised algorithms that exploit the availability of increasingly large volumes of Web pages are learning to extract structured knowledge from unstructured text.

Previous efforts in text-based knowledge acquisition can largely be attributed to the field of Information Extraction (IE), where the task is to recognize entities and relations mentioned within text corpora. Traditional IE systems focused on locating instances of narrow, pre-specified relations, such as the time and place of events, from small, homogeneous corpora. The KNOWITALL system [5] advanced the field of IE by capturing knowledge in a manner that scaled to the size and diversity of relationships present within millions of Web pages. KNOWITALL accomplished this task by learning to label its own training examples using only a set of domain-independent extraction patterns and a bootstrapping procedure. While KNOWITALL is capable of self-supervising its training process, extraction is not fully automatic; KNOWITALL requires a user to name a relation prior to each extraction cycle for every relation of interest. When acquiring knowledge from corpora as

large and varied as the the Web, the task of anticipating all relations of interest becomes highly problematic.

A recent goal for knowledge extraction systems has been to eliminate the need for human involvement, thus making it possible to automate the knowledge acquisition process for a new domain or set of relations. Such efforts have resulted in the paradigm of preemptive or *open* information extraction. Compared to traditional information extraction, open information extraction systems attempt to automatically discover all possible relations from each sentence encountered. Shinyama and Sekine [12] took important steps in the direction of open IE by developing a method for unrestricted relation extraction that could be applied to a small corpus of newswire articles. The TEXTRUNNER system [1] was the first open information extraction system to do this at Web scale, processing just over 110,000,000 Web pages and yielding over 330,000,00 statements about concrete entities with a precision of 88%.

Despite advances in information extraction, the process of text understanding – the formation of a coherent set of beliefs from a textual corpus – involves representation and learning abilities not currently achievable by today's IE systems. While IE systems can uncover assertions about individual objects, a *theory* of a particular domain is built from collective knowledge of concepts and relationships among them. Compared to a vast set of independent statements extracted through IE, a domain theory represents knowledge compactly; relationships are expressed between abstract concepts at an appropriate level of generalization in a hierarchy. Additionally, a domain theory contains only information relevant to the topic at hand.

We believe that theory formation will involve a more complex process in which the corpus is continually analyzed and knowledge is acquired increasingly over time. The amount and richness of information that can be gleaned from large, heterogeneous corpora such as the Web will require an ongoing or *lifelong learning* process in which earlier learned knowledge is used to guide subsequent learning. While open IE further automated the process of relation discovery, theory formation involves the challenge of intelligently mechanizing the exploration of concepts within a given domain.

This paper introduces ALICE,[1] one of the first lifelong learning agents to automatically build a collection of concepts, facts and generalizations about a particular topic directly from a large volume of Web text. Over time, ALICE uses knowledge gained about attributes of the domain to focus its search for additional knowledge.

[1]Not to be confused with the infamous chatbot with the same name, the name ALICE was inspired by the fictional creator of a text-reading agent in Astro Teller's novel, *Exegesis* [15].

In this paper, we:

- Develop the paradigm of *lifelong learning* in the context of hierarchical, unsupervised knowledge acquisition from text.
- Describe ALICE, a lifelong learning agent that builds upon recent advances in unsupervised information extraction to discover new concepts and compose abstract domain knowledge with a precision of 78%.
- Report on a collection of lifelong learning strategies that enable ALICE to iteratively acquire a textual theory at Web scale.

The remainder of the paper is organized as follows. In Section 2, we reintroduce the paradigm of lifelong learning previously articulated by members AI community. In Section 3, we describe ALICE, our embodiment of a lifelong learning agent that builds a domain theory from text. We report on experimental results in Section 4, followed by a review of related research in textual theory composition in Section 5. The paper concludes with a discussion of future work.

2. LIFELONG LEARNING

The paradigm of "*lifelong learning*" was articulated in the machine learning community in response to stark differences observed between human and machine learning abilities [16]. While humans can successfully learn behaviors having had only a small set of marginally related experiences, machine learning algorithms typically have difficulty learning from small datasets. The ability of humans to learn when faced with complex tasks in new settings fraught with rich sensory input can be attributed to the ability to exploit knowledge learned during previous learning tasks. The fact that humans learn continuously – increasingly more complicated concepts and behaviors are developed over the course of an entire lifetime – has motivated the development of bootstrapping algorithms in which knowledge is transferred over time from simple tasks to increasingly more difficult ones.

By definition, lifelong learning agents reduce the difficulty of solving the n^{th} learning problem they face by using knowledge acquired from having solved earlier problems. Thus, the learning process of a lifelong learning agent happens *incrementally*; learning occurs at every time step and knowledge acquired can be used later. Learning also happens *hierarchically*; knowledge is acquired in a bottom-up fashion and can be subsequently built upon and altered. This type of behavior has been implemented in several autonomous agents, mostly with application to robotics [17, 13] and reinforcement-learning-based tasks [10].

An agent that learns incrementally inherently possesses

a mechanism for identifying and executing a series of simple problem-solving tasks from the overall complex problem at hand. This process has historically been cast in terms of one of the most pervasive paradigms of AI – *learning as search*. One of the earliest systems to demonstrate this notion in the area of knowledge representation and discovery was Automated Mathematician (AM) [6]. AM was an agent that attempted to discover new concepts in mathematics and the relationships between them using a heuristic-driven search strategy. While AM notably suffered from several limitations, it demonstrated that open-ended theory formation could be mechanized and modeled as search.

3. LIFELONG KNOWLEDGE ACQUISITION WITH ALICE

This section introduces ALICE, a lifelong learning agent whose goal is to iteratively and hierarchically construct a theory — a collection of concepts, facts and generalizations that describe a particular domain of interest – directly from text. ALICE begins with a domain-specific textual corpus, a source of background knowledge, and a control strategy and embarks on a search to update and refine a theory of the domain. ALICE's domain theory is iteratively updated with various forms of knowledge, including *concepts* and their *instances*, *attributes* of concepts, and the various *relationships* among them. Concepts abstractly refer to all instances of a given category or class of entities (*e.g. orange* is an instance of the concept, <FRUIT>). Attributes of concepts include the various propositions or *relations* in which they take part (*e.g.* a <FRUIT> is something that GROWS). Relationships describe how concepts or instances are associated (*e.g.* GROWIN(<FRUIT>, <LOCATION>)).[2]

ALICE's general learning architecture is depicted in Figure 1. From a given input text corpus T in a domain D, ALICE begins by using the TEXTRUNNER open information extraction system to extract facts about individual objects in the domain from each sentence in the corpus. A fact takes the form of a relational tuple $f = (e_i, r_{i,j}, e_j)$, where e_i and e_j are strings that denote objects in the domain, and $r_{i,j}$ is a string that expresses a relationship believed to exist between them. Each tuple is assigned a probability based on a probabilistic model of redundancy in text [3].

The set of individual assertions output by TEXTRUNNER serves as the basis from which ALICE then proceeds to add general knowledge to its domain theory. Beginning with a single domain-specific concept c_0, which can be specified manually or heuristically, ALICE uses a set of learners L and a given search strategy S to develop an agenda A of learning tasks to be completed over time.

[2]A domain theory may also contain rules describing dependencies among propositions (*e.g.* GROWIN(X, Y) ∧ ISA(X, <CITRUS FRUIT>) → HASWARMCLIMATE(Y)). We anticipate adding the ability to learn rules in future work.

In addition to the TEXTRUNNER system that provides ALICE with the ability to uncover a set of specific facts about the domain, ALICE's set of learners currently includes modules for *concept discovery* and *proposition abstraction*.

Each of ALICE's learning tasks is defined by a concept c and a set of attributes $\pi(c)$. Attributes take the form of relations associated with the current concept according to ALICE's unrestricted relation extraction process. In our current implementation, relations in $\pi(c)$ are explored in descending order of the frequency with which they have been observed to occur in the corpus along with a sample of known instances of the concept c. Although π is currently implemented as a heuristic function, we aim to have ALICE learn π as well. We describe ALICE's learning process in more detail in Section 3.1.

The final property that describes ALICE's behavior is S, the strategy that defines the order in which individual learning tasks are addressed. Instead of taking an uninformed, exhaustive path to theory construction, ALICE is guided by knowledge accrued about attributes perceived to be highly relevant to understanding of the domain. One strategy may be for the system to thoroughly explore properties of the first concept it discovers before attempting to learn about other concepts encountered subsequently. Alternatively, one can imagine an agent in possession of a short attention span who upon discovery of a new concept, immediately shifts its current focus to learn something about the new item. Thus, S can be exhaustive (*e.g.* depth-first or breadth-first search) or heuristic-driven (*e.g.* best-first or A^* search). We describe several lifelong search strategies in Section 3.2.

The knowledge output upon completion of a learning task is used in two ways: to update the current domain theory and to generate subsequent learning tasks. This very behavior is what makes ALICE a lifelong agent – ALICE uses the knowledge acquired during the n^{th} learning task to specify its future learning agenda. ALICE's general learning behavior is summarized in Figure 2.

3.1 Theory Learning

For our experiments, we have given ALICE access to 2.5 million Web pages focused on the topic of nutrition. As an initial source of knowledge, ALICE makes use of WordNet, a hand-built semantic lexicon for the English language that groups 117,097 distinct nouns into 81,426 concepts.[3] WordNet also expresses a handful of semantic relationships between concepts, such as hypernymy/hyponymy and holonymy/meronymy. Despite its ability to cover a wide range of concepts, this knowledge

[3]Although ALICE receives a background theory as input, the system learns to add knowledge without hand-tagged examples or manual intervention, and is thus unsupervised.

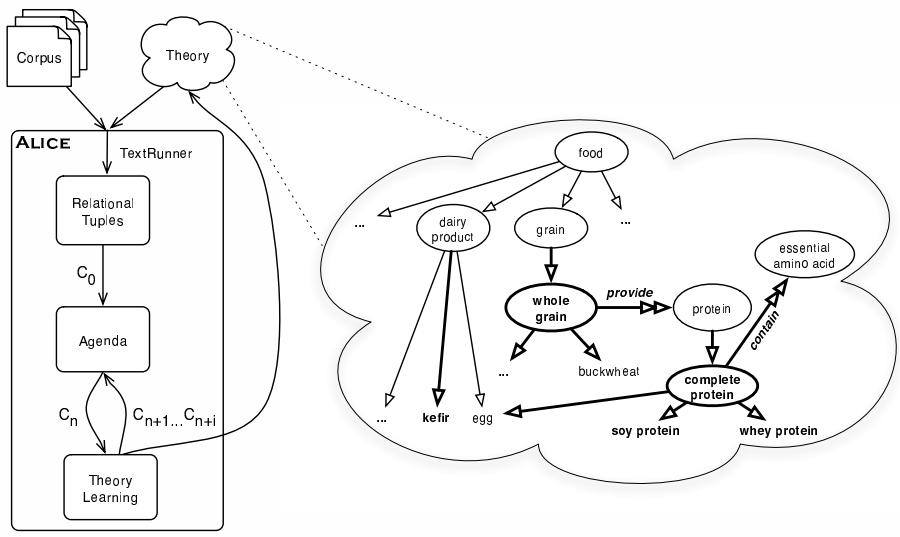

Figure 1: System Overview of Alice. Given a domain, corpus and background theory, Alice iteratively adds general, domain knowledge to its theory. The output of each learning cycle suggests the focus of subsequent learning tasks. Examples of the concepts, instances and relations automatically added to Alice's theory are given in boldface.

source falls short in its ability to offer complete coverage of the domain. WordNet's knowledge of entities and concepts is incomplete, and it fails to specify the myriad of relationships that exist between them. For example, in the nutrition domain, we observed omissions such as the lack of an entry for the *noni*, an exotic fruit touted for its potent antioxidant properties, no concept of "*healthy food*" and the inability to express the belief that foods rich in antioxidants may prevent disease if present in one's diet.

3.1.1 Concept Discovery

Although the initial theory provided by WordNet contains a large number of entities and concepts, it is by no means exhaustive. Therefore, in addition to the hypernymy and hyponymy information available in WordNet, ALICE acquires class membership information directly from its corpus using a set of domain-independent extraction patterns and probabilistic assessment model employed by KNOWITALL. This data-driven approach makes it possible to acquire knowledge about entities missing from WordNet with high accuracy.

At the beginning of each learning task, ALICE receives an unexplored concept c_n as input, instantiates the set of extraction patterns (*e.g.* <x> *such as* <y>) with the name of c_n, (*e.g. fruit such as* <y>), applies them over the input corpus, and uses the assessment model to add high-probability instances of c_n to its theory.

ALICE's initial theory contains useful knowledge in the form of previously discovered instances and some classes to which they belong, such as the knowledge that *buckwheat* is a type of *grain* and that *grain* is a type of *food*. Yet the concept hierarchy is in some places incomplete. ALICE further utilizes the small set of extraction patterns to add more fine-grained classes of existing concepts present in its theory. By applying the constrained extraction patterns using our knowledge of *buckwheat*, (*e.g.* "*<? grains> such as buckwheat*", "*buckwheat is a < ? food >*") and using a redundancy-based model of fact assessment over ALICE's corpus, ALICE finds that *buckwheat* is not just any *grain* or *food*, but more specifically, a type of *whole grain, gluten-free grain, fiber-rich food* and *nutritious food*.

Occasionally we find that an existing WordNet concept possesses more than one *sense*. While *fruit* is most often used to refer to a ripe object produced by a seed plant, it can also designate the consequence of some action. While it may be possible to disambiguate among various senses of a word, we found that using the heuristic of placing the newly discovered subcategory under the most frequent sense of the word according to WordNet served well.

Upon inspection of approximately 100 random samples of subclasses proposed by this method, we found that 78.3% of the classes were legitimate and meaningful additions to the theory (*e.g. oily fish, leafy green vegetable, complex carbohydrate*), 14.1% were vacuous or not immediately useful (*e.g. important antioxidant, favorite food*), and the remaining 7.6% were errors.

3.1.2 Proposition Abstraction

Another type of information that can be added to ALICE's domain theory are *general propositions* found by the process of *abstraction*. Compared to statements about individual entities found by relation extraction in TEXTRUNNER, general propositions express relationships among concepts. Such relationships encode "reasonable general claims" about the world from particular facts [11]. In our domain, the general proposition PROVIDE(<FRUIT>, <VITAMIN>) should be deduced from the set of tuples { (*oranges, provide, vitamin c*), (*bananas, provide, a source of B vitamins*), (*an avocado, provides, niacin*)}. A key challenge in deducing general claims from a set of isolated facts is finding a suitable level of generalization in the concept hierarchy. Consider for example, the difference in knowing PROVIDE(<FRUIT>, <VITAMIN>) versus PROVIDE(<FOOD>, <SUBSTANCE>). The goal of proposition abstraction is to find the lowest point in concept hierarchy that describes a set of related instances.

The task of proposition abstraction can also be cast as the process of estimating the likelihood that a given concept appears as an argument to a given relation. Yet relations often take multiple concepts as arguments – the predicate PROVIDE(X,Y) can describe both a relationship between <FRUIT> and <VITAMIN> as well as one between <COMPANY> and <PRODUCT>. If in addition to the three tuples observed in the previous example, we saw that (*Vitamin World, provides, Super Antioxidant Plus*), a naive approach might propose PROVIDE(<ENTITY>, <ENTITY>), an abstraction too general to be of use. We use this observation to motivate a clustering-based approach to proposition abstraction, which we now describe.

Given an input relation r whose arguments we wish to generalize, ALICE obtains a set of TEXTRUNNER tuples t that match the form (e_i, r, e_j). The system then examines each entity e_i and e_j in t and tries to obtain the set of concepts to which they belong. In some cases, ALICE will already have this information in the theory; otherwise, ALICE will take some time to learn about the possible class and subclass memberships for a new instance using the discovery process described in the previous section. The system automatically clusters the tuples in t , using class-membership features where known, and words frequently observed to appear in the individual entity strings. For each cluster $t_k \in t$ ALICE generates a set of abstractions using the following algorithm, and adds them to the theory.

Given a cluster of tuples, t_k, ALICE obtains the set of entities $e_{i,1} \ldots e_{i,|t_k|}$ observed in the i^{th} argument position and retrieves the set of concepts that possibly characterize it according to the current theory. The system then computes an association measure M that measures the likelihood that each concept c describes

```
ALICE ( Corpus T,
        Background Theory θ,
        Search Strategy S )
  Facts F ← TEXTRUNNER(T)
  Agenda A ← {c_0}
  For n = 0 ... ∞
    c_n ← next(A)
    If IsNewConcept(c_n)
      θ' ← DiscoverConcept(c_n, T, θ)
      θ ← θ'
    r = SelectAttribute(c_n)
    θ' ← Abstract(r, c_n, F, θ)
    A ← Insert(A, c_{n+1} ... c_{n+i}) according to S
    θ ← θ'
```

Figure 2: Alice's Lifelong Learning Process

the i^{th} argument slot in the context of the relation r:

$$M(arg_i = c|r) = w_c\Big(\frac{\sum_{j=1}^{|T_k|} Count(r, e_{i,j} \in c)}{|t_k|}\Big)$$

Since we typically prefer more specific concepts to those that are more general, w_C weights each concept according to its place in the theory's current concept hierarchy, biasing the measure by $\frac{1}{d}$, where d is the maximum distance found between any entity $e_{i,j}$ and the concept c under consideration. The algorithm greedily searches for the combination of conceptual slot descriptions that best covers the set of tuples in the cluster, with the constraint that an abstraction must cover a minimum number of tuples to be meaningful. If any tuples remain undescribed by the best abstraction , the procedure is repeated until all items in the cluster are fully covered, or until we can propose no more generalizations.

In order to sidestep the problem of needing to prespecify the number of clusters to find in the data, we first employed a expectation-maximization clustering algorithm that used cross validation to determine the number of clusters. We found the approach to not only be too slow in practice, but unable to produce clusters that were sufficiently fine-grained. We found a better solution in k-means clustering [8]. Although k-means requires the number of clusters to be specified *a priori*, the system used its theory to estimate k as the maximum number of distinct concepts present in each argument position according to the entities in the cluster. In practice, this approach results in a large number of clusters, which can easily be reduced via a post-processing step that merges clusters whose centroids were found to be close together. We empirically evaluate ALICE's ability to generalize from individual facts in Section 4.

3.2 Search

We now describe how ALICE generates and prioritizes subsequent theory-learning tasks. Recall that our goal is to automatically construct a theory of a particular topic from a textual corpus. In some cases, such as when working with a set of automatically gathered Web pages or a general-purpose text collection, out-of-domain information may be present. A page might mention the nutrient *iron* as part of a discussion about health conditions in which the body is unable to absorb it. A text collection might contain pages about an object in the domain in a different sense of the word (*e.g. iron*, a device used to press clothing). Therefore, ALICE's search must be guided towards domain-specific concepts and relations as much as possible. An uninformed exhaustive strategy that simply iterates over all concepts and relations found in the input corpus independent of existing domain knowledge will likely introduce spurious information into our domain theory.

3.2.1 Best-first search

Our first attempt at building a lifelong learner produced a heuristic-driven agent who is eager to explore new concepts at every opportunity. For instance, after beginning with the concept <FRUIT> and learning that in general, the proposition CONTAIN(<FRUIT>, <ANTIOXIDANT>) holds, ALICE is eager to learn more about the concept of <ANTIOXIDANT>. This inclination feels quite natural. Yet upon subsequently learning that MAYPREVENT(<ANTIOXIDANT>, <DISEASE>) is true in the general case, ALICE prefers to see what can be gleaned by learning about diseases. This greedy approach can sometimes send ALICE into a tailspin.

Formally, after an initial concept c_0 has been specified to ALICE, the agent executes $\pi(c_0)$ to obtain the next relation r associated with the concept. ALICE uses the set of learners to add general propositions of r and further develop concepts encountered during this procedure. After completing the n^{th} learning cycle, all concepts c' which have been activated while learning about r are put into A, which in this case, is a priority queue. During best-first search, the concepts in A are ordered in descending order according to how many instances of the concept have been newly discovered by the learner. Other possible heuristics include an ordering of concepts based on distance in the concept hierarchy, or the similarity with which attributes have been observed to occur for concepts under consideration.

3.2.2 Associative search

We developed an alternate search strategy based on yet another natural learning process – search by analogy, or what we will refer to as *associative search*. Broadly speaking, learning by analogy utilizes the structure and outcome of the agent's problem-solving history to learn something about a new problem.

During associative search, ALICE uses the output of iteration n to immediately conjecture similar learning tasks for iteration $n+1$. For example, if ALICE learns that CONTAIN(<FRUIT>, <VITAMIN>), it reasonable to expect a cohesive theory to tell us what, if anything else in the domain, contains vitamins. Associative search provides an elegant manner for exploring related concepts while remaining focused on the domain at hand. If as before, ALICE concluded that MAYPREVENT(<ANTIOXIDANT>, <DISEASE>), the agent will next try to understand what else may prevent disease, rather than fall prey to exploring a tangent on the topic of disease that gradually shifts its focus away from the domain of interest.

Upon learning about an active concept c_n having attribute r, associative search places all c' found to be related to c_n via r into the concept list A with attribute r. After ALICE has completed this stack of learning tasks, (A is empty), a new attribute of c_n is chosen by the function π, using the frequency-based ordering described previously.

3.2.3 Breadth-Limited search

The final search strategy we implemented was a simple one that encouraged the agent to explore concepts in a semi-exhaustive fashion. Instead of pursuing a fully exhaustive strategy in which the agent attempts to learn a complete set of statements about a concept before moving on to another, we implemented breadth-first search algorithm that limits the number of concepts explored based on information obtained during previous learning tasks.

From an implementation standpoint, this search strategy is identical to the best-first strategy with one key exception – concepts placed in A are explored in order of when they enter the queue, rather than sorted by the "newness" criteria previously defined. Each time a concept c' is activated during a learning task involving c_n at depth d, a new learning task for c' is generated for depth $d+1$. While ALICE might end up learning some additional facts about <DISEASE>, this will happen incrementally at an appropriate rate relative to other concepts, rather than allowing the agent to spend all immediate resources learning about disease and related concepts.

4. EXPERIMENTAL RESULTS

In this section, we empirically assess ALICE's ability to add a collection of abstract assertions about nutrition from a 2.5-million-page Web crawl constructed to contain documents in the domain. There are several criteria at play when judging the overall "goodness" of the learned abstractions. Our assessment should take into account an abstraction's ability to generalize from instances to concepts at an appropriate level. Our evaluation metric should also consider whether or not the

abstraction in question is on topic given the domain. This criteria provides us with a way to assess the performance of each search strategy we have implemented.

We ran each search strategy over the same corpus for a fixed number of learning steps (200), and chose a random sample of 200 abstract propositions for each of the three search methods. Using a modified version of an assessment scheme previously developed by Schubert [11], one of the authors of this paper was asked to assign each proposition P one of the following labels in the context of the domain D:

Off-Topic: P expresses a fact not central to D,
e.g. CAUSE(<ORGANISM>, <DISEASE>)
True: P is a reasonable general claim about D,
e.g. PROVIDE(<FRUIT>, <VITAMIN>)
Vacuous: One or more arguments of P are not appropriately specific/general,
e.g. PROVIDE(<FOOD>, <SUBSTANCE>),
Incomplete: P is missing something,
e.g. PROVIDE(<ANTIOXIDANT>, <BODY PART>)
Error: P contradicts our knowledge of D,
e.g. BENOT(<FRUIT>, <FOOD>)

The results of our assessment are given in Table 1. The heuristic-driven best-first search was observed to venture off-topic after a few iterations, yielding a theory judged to contain a sizable amount of out-of-domain knowledge. Once ALICE has discovered the concept of <SYMPTOM>, ALICE pursues a set of illness-related concepts including <DISEASE>, <INFECTION> among others. This failure can be attributed not only to the greediness of the heuristic and the imperfect nature of the Web corpus from which ALICE constructs the theory, but by the overly general concept, <FOOD>, that generated this particular learning task. The fact that the concept of <FOOD> is positioned fairly high in the concept hierarchy has the impact that it stands in relation to quite a number of concepts, a proportion of which will be out-of-domain.

Compared to best-first search, associative and breadth-limited search fared much better, adding appropriately-general, domain-specific abstractions to ALICE's theory with a precision of 78.0% and 75.5%, respectively. The bulk of the remaining assertions were characterized as on-topic but vacuously true — 9.5% in the case of associative search, and 12.5% in the case of breadth-limited search. The small amount of propositions judged as "incomplete" is due to a limitation of the knowledge representation scheme in TEXTRUNNER in which n-ary relations are currently not handled. The few instances deemed to be "errors" were largely due to cascading errors originating with TEXTRUNNER or an inability to correctly disambiguate objects having multiple senses.

We also compared the recall of the three approaches, as shown in Figure 3. Since true recall cannot be easily computed from an unstructured Web corpus, we measure recall in terms of the size of the set of distinct abstractions proposed. Although the best-first strategy outputs a large number of generalizations, recall that a low proportion of them were judged to meet our measure of goodness. Using our precision assessment, both associative and breadth-limited search are estimated to output a greater proportion of correct, on-topic generalizations after 200 iterations – 543/696 and 779/1038 respectively – compared to only 517/1309 for best-first search.

	On-Topic				Off
	True	Vacuous	Inc	Error	Topic
Best-First	39.5%	6.5%	0.0%	1.0%	43.0%
Associative	**78.0%**	9.5%	3.0%	3.5%	6.0%
Breadth-Limited	75.5%	12.5%	3.0%	1.5%	7.5%

Table 1: Precision of Proposition Abstraction

Finally, we measured the diversity of concepts and relations expressed within the propositions found by each search algorithm, and found that the statements cover a wide variety of attributes of the domain. Associative search was found to propose statements about distinct concepts at a rate of 0.67, and novel predicates at a rate of 0.32. The propositions output by best-first search involved a large number of distinct concepts (0.78), but a smaller number of predicates (0.23). Breadth-limited search proposed abstractions at a rate that contains the fewest number of distinct concepts (0.56), while the rate of unique predicates was observed to be 0.29. Nearly all of the concept-to-concept relationships proposed by ALICE were novel relative to the input source of WordNet – 98.5% of the propositions output by assocative search were not already present in the background onotlogy. Similar results were observed for breadth-limited search (99.5%) and best-first search (95.5%).

5. RELATED WORK

To date, the task of inducing domain theories directly from textual corpora has been explored by only a few systems. Liakata and Pulman [7] showed that they could induce a logically structured inference rules from a 4000-word corpus describing company succession events. While the method was anecdotally shown to learn a handful of useful domain-specific rules, to our knowledge, an extensive empirical evaluation of the system output has not been reported.

While previous efforts to augment the WordNet ontology are too numerous to discuss here, most recently, Suchanek *et. al.* developed YAGO [14], a system capable of unifying facts automatically extracted from Wikipedia Web pages to concepts in WordNet. While YAGO performs this unification with an accuracy of

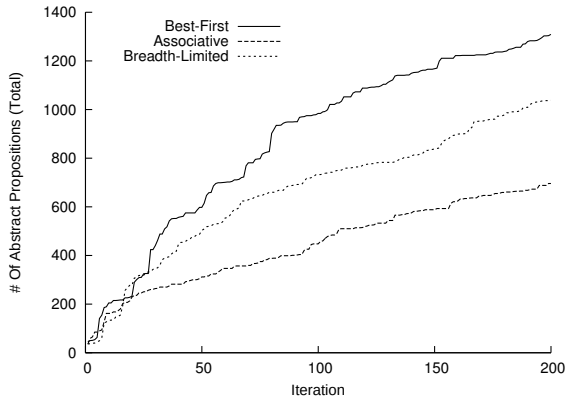

Figure 3: Knowledge Acquisition By Search Strategy. For each algorithm, recall is computed as the total number of distinct abstractions proposed after each search iteration.

95%, its coverage is currently limited to a handful of relations that are derivable using the structure and headings contained within Wikipedia pages.

Earlier, Schubert and Tong developed a method for extracting "general world knowledge" of the flavor deduced by ALICE. Their approach took a set of human-validated parse trees from the Penn Treebank, and used a combination of tree-based pattern matching and heuristics to obtain a set of abstract propositions. Using similar criteria that we have employed in our evaluation of ALICE, human assessors found about 60% of statements extracted by this method to be "reasonable general claims." Also related to the task of deriving relationships between concepts is Clark and Weir's use of parsed corpora and chi-squared statistics to induce relationships between existing concepts in WordNet [2]. Their method was evaluated indirectly by its ability to improve a natural language disambiguation task on which a precision of around 75% was obtained.

While these methods attempt to solve the same task of finding abstract propositions within text, their inherent assumptions – the use of small, parsed corpora – make it impossible to perform a direct comparison relative to the method we have presented in this paper. The success of ALICE relies not on the ability to parse its input corpus, which is orders of magnitude larger, but on unsupervised techniques that exploit the redundancy of information found within unstructured text.

6. CONCLUSIONS AND FUTURE WORK

We have introduced ALICE, one of the first lifelong learning agents capable of building a domain theory from a large collection of Web text. ALICE uses information from previous knowledge acquisition tasks to iteratively compose individual statements output by a state-of-the-art information extraction system into an abstract domain theory with a precision of 78%.

In the immediate future, we plan to explore the notion of mutual recursion in the context of ALICE. Mutual recursion can take several forms, including the use of the domain theory output by ALICE to improve modules like TEXTRUNNER that underlie its learning process, and the construction of inference rules from ALICE's general world knowledge. Finally, we plan to explore what ALICE can learn from its search through the space of knowledge, making it possible to both eliminate fruitless learning tasks and identify areas in which ALICE's theory can be further enriched.

7. ACKNOWLEDGMENTS

This research was supported in part by NSF grants IIS-0535284 and IIS-0312988, DARPA contract NBCHD030010, ONR grant N00014-05-1-0185 as well as gifts from Google, and carried out at the University of Washington's Turing Center.

8. REFERENCES

[1] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Procs. of IJCAI*, 2007.

[2] S. Clark and D. Weir. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2), 2002.

[3] D. Downey, O. Etzioni, and S. Soderland. A probabilistic model of redundancy in information extraction. In *Procs. of IJCAI 2005*, 2005.

[4] O. Etzioni, M. Banko, and M. Cafarella. Machine reading. In *AAAI*, 2006.

[5] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[6] D. Lenat. Automated theory formation in mathematics. In *Procs. of IJCAI*, 1977.

[7] M. Liakata and S. Pulman. Learning theories from text. In *Procs. of COLING*, 2004.

[8] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[9] T. Mitchell. Reading the web: A breakthrough goal for AI. In *AI Magazine*. AAAI Press, 2005.

[10] M. B. Ring. CHILD: A first step towards continual learning. *Machine Learning*, 28:77–105, 1997.

[11] L. Schubert and M. Tong. Extracting and evaluating general world knowledge from the brown corpus. In *Proc. of the HLT/NAACL Workshop on Text Meaning*, 2003.

[12] Y. Shinyama and S. Sekine. Preemptive information extraction using unrestricted relation discovery. In *Procs. of HLT/NAACL*, 2006.

[13] P. Stone and M. Veloso. Layered learning. In *Proc. of ECML*, 2000.

[14] F. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *Procs. of WWW*, 2007.

[15] A. Teller. *Exegesis*. Random House, 1999.

[16] S. Thrun. Lifelong learning algorithms. In S. Thrun and L. Pratt, editors, *Learning To Learn*. Kluwer Academic Publishers, 1998.

[17] S. Thrun and T. Mitchell. Lifelong robot learning. *Robotics and Autonomous Systems*, 15:25–46, 1995.