Toward an Architecture for Never-Ending Language Learning

Andrew Carlson[1], **Justin Betteridge**[1], **Bryan Kisiel**[1], **Burr Settles**[1],
Estevam R. Hruschka Jr.[2], and **Tom M. Mitchell**[1]
[1]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]Federal University of São Carlos, São Carlos, SP, Brazil

Abstract

We consider here the problem of building a never-ending language learner; that is, an intelligent computer agent that runs forever and that each day must (1) extract, or read, information from the web to populate a growing structured knowledge base, and (2) learn to perform this task better than on the previous day. In particular, we propose an approach and a set of design principles for such an agent, describe a partial implementation of such a system that has already learned to extract a knowledge base containing over 242,000 beliefs with an estimated precision of 74% after running for 67 days, and discuss lessons learned from this preliminary attempt to build a never-ending learning agent.

Introduction

We describe here progress toward our longer-term goal of producing a never-ending language learner. By a "never-ending language learner" we mean a computer system that runs 24 hours per day, 7 days per week, forever, performing two tasks each day:

1. *Reading task:* extract information from web text to further populate a growing knowledge base of structured facts and knowledge.
2. *Learning task:* learn to read better each day than the day before, as evidenced by its ability to go back to yesterday's text sources and extract more information more accurately.

The thesis underlying this research is that the vast redundancy of information on the web (e.g., many facts are stated multiple times in different ways) will enable a system with the right learning mechanisms to succeed. One view of this research is that it is a case study in lifelong, or never-ending learning. A second view is that it is an attempt to advance the state of the art of natural language processing. A third view is that it is an attempt to develop the world's largest structured knowledge base – one that reflects the factual content of the world wide web, and that would be useful to many AI efforts.

In this paper, we first describe a general approach to building a never-ending language learner that uses semi-supervised learning methods, an ensemble of varied knowledge extraction methods, and a flexible knowledge base representation that allows the integration of the outputs of those methods. We also discuss design principles for implementing this approach.

We then describe a prototype implementation of our approach, called *Never-Ending Language Learner* (NELL). At present, NELL acquires two types of knowledge: (1) knowledge about which noun phrases refer to which specified semantic *categories*, such as cities, companies, and sports teams, and (2) knowledge about which pairs of noun phrases satisfy which specified semantic *relations*, such as hasOfficesIn(organization, location). NELL learns to acquire these two types of knowledge in a variety of ways. It learns free-form text patterns for extracting this knowledge from sentences on the web, it learns to extract this knowledge from semi-structured web data such as tables and lists, it learns morphological regularities of instances of categories, and it learns probabilistic horn clause rules that enable it to infer new instances of relations from other relation instances that it has already learned.

Finally, we present experiments showing that our implementation of NELL, given an initial seed ontology defining 123 categories and 55 relations and left to run for 67 days, populates this ontology with 242,453 new facts with estimated precision of 74%.

The main contributions of this work are: (1) progress toward an architecture for building a never-ending learning agent, and a set of design principles that help successfully implement that architecture, (2) a web-scale experimental evaluation of an implementation of that architecture, and (3) one of the largest and most successful implementations of bootstrap learning to date.

Approach

Our approach is organized around a shared knowledge base (KB) that is continuously grown and used by a collection of learning/reading subsystem components that implement complementary knowledge extraction methods. The starting KB defines an ontology (a collection of predicates defining categories and relations), and a handful of seed examples for each predicate in this ontology (e.g., a dozen example cities). The goal of our approach is to continuously grow this KB by reading, and to learn to read better.

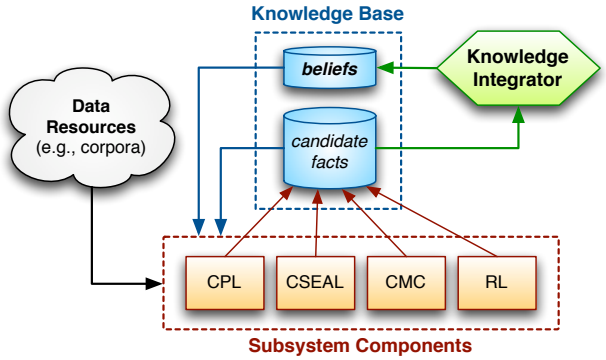

Figure 1: Our Never-Ending Language Learner (NELL) architecture. See "Approach" for an overview of the approach implemented in NELL, and "Implementation" for subsystem details.

Category and relation instances added to the KB are partitioned into *candidate facts* and *beliefs*. The subsystem components can read from the KB and consult other external resources (e.g., text corpora or the Internet), and then propose new candidate facts. Components supply a probability for each proposed candidate and a summary of the source evidence supporting it. The Knowledge Integrator (KI) examines these proposed candidate facts and promotes the most strongly supported of these to belief status. This flow of processing is depicted in Figure 1.

In our initial implementation, this loop operates iteratively. On each iteration, each subsystem component is run to completion given the current KB, and then the KI makes decisions on which newly proposed candidate facts to promote. The KB grows iteration by iteration, providing more and more beliefs that are then used by each subsystem component to retrain itself to learn to read better on the next iteration. In this way, our approach can be seen as implementing a coupled, semi-supervised learning method in which multiple components learn and share complementary types of knowledge, overseen by the KI. One can view this approach as an approximation to an Expectation Maximization (EM) algorithm in which the E step involves iteratively estimating the truth values for a very large set of virtual candidate beliefs in the shared KB, and the M step involves retraining the various subsystem component extraction methods.

This kind of iterative learning approach can suffer if labeling errors accumulate. To help mitigate this issue, we will allow the system to interact with a human for 10–15 minutes each day, to help it stay "on track." However, in the work reported here, we make limited use of human input.

The following design principles are important in implementing our approach:

- Use subsystem components that make uncorrelated errors. When multiple components make uncorrelated errors, the probability that they all make the same error is the product of their individual error probabilities, resulting in much lower error rates.
- Learn multiple types of inter-related knowledge. For example, we use one component that learns to extract predicate instances from text resources, and another which learns to infer relation instances from other beliefs in the KB. This provides multiple, independent sources of the same types of beliefs.
- Use coupled semi-supervised learning methods to leverage constraints between predicates being learned (Carlson et al. 2010). To provide opportunities for coupling, arrange categories and relations into a taxonomy that defines which categories are subsets of which others, and which pairs of categories are mutually exclusive. Additionally, specify the expected category of each relation argument to enable type-checking. Subsystem components and the KI can benefit from methods that leverage coupling.
- Distinguish high-confidence beliefs in the KB from lower-confidence candidates, and retain source justifications for each belief.
- Use a uniform KB representation to capture candidate facts and promoted beliefs of all types, and use associated inference and learning mechanisms that can operate on this shared representation.

Related Work

AI has a long history of research on autonomous agents, problem solving, and learning, e.g., SOAR (Laird, Newell, and Rosenbloom 1987), PRODIGY (Carbonell et al. 1991), EURISKO (Lenat 1983), ACT-R (Anderson et al. 2004), and ICARUS (Langley et al. 1991). In comparison, our focus to date has been on semi-supervised learning to read, with less focus on problem-solving search. Nevertheless, earlier work provides a variety of design principles upon which we have drawn. For example, the role of the KB in our approach is similar to the role of the "blackboard" in early systems for speech recognition (Erman et al. 1980), and the frame-based representation of our KB is a reimplementation of the THEO system (Mitchell et al. 1991) which was originally designed to support integrated representation, inference and learning.

There is also previous research on life-long learning, such as Thrun and Mitchell (1995), which focuses on using previously learned functions (e.g., a robot's next-state function) to bias learning of new functions (e.g., the robot's control function). Banko and Etzioni (2007) consider a lifelong learning setting where an agent builds a *theory* of a domain, and explore different strategies for deciding which of many possible learning tasks to tackle next. Although our current system uses a simpler strategy of training all functions on each iteration, choosing what to learn next is an important capability for lifelong learning.

Our approach employs semi-supervised bootstrap learning methods, which begin with a small set of labeled data, train a model, then use that model to label more data. Yarowsky (1995) uses bootstrap learning to train classifiers for word sense disambiguation. Bootstrap learning has also been employed successfully in many applications, including web page classification (Blum and Mitchell 1998), and named entity classification (Collins and Singer 1999).

Bootstrap learning approaches can often suffer from "semantic drift," where labeling errors in the learning process can accumulate (Riloff and Jones 1999; Curran, Murphy, and Scholz 2007). There is evidence that constraining the learning process helps to mitigate this issue. For example, if classes are mutually exclusive, they can provide negative examples for each other (Yangarber 2003). Relation arguments can also be type-checked to ensure that they match expected types (Paşca et al. 2006). Carlson et al. (2010) employ such strategies and use multiple extraction methods, which are required to agree. Carlson et al. refer to the idea of adding many constraints between functions being learned as "coupled semi-supervised learning." Chang, Ratinov, and Roth (2007) also showed that enforcing constraints given as domain knowledge can improve semi-supervised learning.

Pennacchiotti and Pantel (2009) present a framework for combining the outputs of an ensemble of extraction methods, which they call "Ensemble Semantics." Multiple extraction systems provide candidate category instances, which are then ranked using a learned function that uses features from many different sources (e.g., query logs, Wikipedia). Their approach uses a more sophisticated ranking method than ours, but is not iterative. Thus, their ideas are complementary to our work, as we could use their ranking method as part of our general approach.

Other previous work has demonstrated that pattern-based and list-based extraction methods can be combined in a synergistic fashion to achieve significant improvements in recall (Etzioni et al. 2004). Downey, Etzioni, and Soderland (2005) presented a probabilistic model for using and training multiple extractors where the extractors (in their work, different extraction patterns) make uncorrelated errors. It would be interesting to apply a similar probabilistic model to cover the setting in this paper, where there are multiple extraction methods which themselves employ multiple extractors (e.g., textual patterns, wrappers, rules).

Nahm and Mooney (2000) first demonstrated that inference rules could be mined from beliefs extracted from text.

Our work can also be seen as an example of multi-task learning in which several different functions are trained together, as in (Caruana 1997; Yang, Kim, and Xing 2009), in order to improve learning accuracy. Our approach involves a kind of multi-task learning of multiple types of functions — 531 functions in total in the experiments reported here — in which different methods learn different functions with overlapping inputs and outputs, and where constraints provided by the ontology (e.g., 'athlete' is a subset of 'person', and mutually exclusive with 'city') support accurate semi-supervised learning of the entire ensemble of functions.

Implementation

We have implemented a preliminary version of our approach. We call this implementation *Never-Ending Language Learner* (NELL). NELL uses four subsystem components (Figure 1):

- *Coupled Pattern Learner* (CPL): A free-text extractor which learns and uses contextual patterns like "mayor of X" and "X plays for Y" to extract instances of categories and relations. CPL uses co-occurrence statistics between noun phrases and contextual patterns (both defined using part-of-speech tag sequences) to learn extraction patterns for each predicate of interest and then uses those patterns to find additional instances of each predicate. Relationships between predicates are used to filter out patterns that are too general. CPL is described in detail by Carlson et al. (2010). Probabilities of candidate instances extracted by CPL are heuristically assigned using the formula $1-0.5^c$, where c is the number of promoted patterns that extract a candidate. In our experiments, CPL was given as input a corpus of 2 billion sentences, which was generated by using the OpenNLP package[1] to extract, tokenize, and POS-tag sentences from the 500 million web page English portion of the ClueWeb09 data set (Callan and Hoy 2009).
- *Coupled SEAL* (CSEAL): A semi-structured extractor which queries the Internet with sets of beliefs from each category or relation, and then mines lists and tables to extract novel instances of the corresponding predicate. CSEAL uses mutual exclusion relationships to provide negative examples, which are used to filter out overly general lists and tables. CSEAL is also described by Carlson et al. (2010), and is based on code provided by Wang and Cohen (2009). Given a set of seed instances, CSEAL performs queries by sub-sampling beliefs from the KB and using these sampled seeds in a query. CSEAL was configured to issue 5 queries for each category and 10 queries for each relation, and to fetch 50 web pages per query. Candidate facts extracted by CSEAL are assigned probabilities using the same method as for CPL, except that c is the number of unfiltered wrappers that extract an instance.
- *Coupled Morphological Classifier* (CMC): A set of binary L_2-regularized logistic regression models—one per category—which classify noun phrases based on various morphological features (words, capitalization, affixes, parts-of-speech, etc.). Beliefs from the KB are used as training instances, but at each iteration CMC is restricted to predicates which have at least 100 promoted instances. As with CSEAL, mutual exclusion relationships are used to identify negative instances. CMC examines candidate facts proposed by other components, and classifies up to 30 new beliefs per predicate per iteration, with a minimum posterior probability of 0.75. These heuristic measures help to ensure high precision.
- *Rule Learner* (RL): A first-order relational learning algorithm similar to FOIL (Quinlan and Cameron-Jones 1993), which learns probabilistic Horn clauses. These learned rules are used to infer new relation instances from other relation instances that are already in the KB.

Our implementation of the Knowledge Integrator (KI) promotes candidate facts to the status of beliefs using a hard-coded, intuitive strategy. Candidate facts that have high confidence from a single source (those with posterior > 0.9) are promoted, and lower-confidence candidates are promoted if

[1] `http://opennlp.sourceforge.net`

they have been proposed by multiple sources. KI exploits relationships between predicates by respecting mutual exclusion and type checking information. In particular, candidate category instances are not promoted if they already belong to a mutually exclusive category, and relation instances are not promoted unless their arguments are at least candidates for the appropriate category types (and are not already believed to be instances of a category that is mutually exclusive with the appropriate type). In our current implementation, once a candidate fact is promoted as a belief, it is never demoted. The KI is configured to promote up to 250 instances per predicate per iteration, but this threshold was rarely hit in our experiments.

The KB in NELL is a reimplementation of the THEO frame-based representation (Mitchell et al. 1991) based on Tokyo Cabinet[2], a fast, lightweight key/value store. The KB can handle many millions of values on a single machine.

Experimental Evaluation

We conducted an experimental evaluation to explore the following questions:

- Can NELL learn to populate many different categories (100+) and relations (50+) for dozens of iterations of learning and maintain high precision?
- How much do the different components contribute to the promoted beliefs held by NELL?

Methodology

The input ontology used in our experiments included 123 categories each with 10–15 seed instances and 5 seed patterns for CPL (derived from Hearst patterns (Hearst 1992)). Categories included locations (e.g., mountains, lakes, cities, museums), people (e.g., scientists, writers, politicians, musicians), animals (e.g., reptiles, birds, mammals), organizations (e.g., companies, universities, web sites, sports teams), and others. 55 relations were included, also with 10–15 seed instances and 5 negative instances each (which were typically generated by permuting the arguments of seed instances). Relations captured relationships between the different categories (e.g., teamPlaysSport, bookWriter, companyProducesProduct).

In our experiments, CPL, CSEAL, and CMC ran once per iteration. RL was run after each batch of 10 iterations, and the proposed output rules were filtered by a human. Manual approval of these rules took only a few minutes.

To estimate the precision of the beliefs in the KB produced by NELL, beliefs from the final KB were randomly sampled and evaluated by several human judges. Cases of disagreement were discussed in detail before a decision was made. Facts which were once true but are not currently (e.g., a former coach of a sports team) were considered to be correct for this evaluation, as NELL does not currently deal with temporal scope in its beliefs. Spurious adjectives (e.g., "today's Chicago Tribune") were allowed, but rare.

[2] http://1978th.net/tokyocabinet

Predicate	Instance	Source(s)
ethnicGroup	Cubans	CSEAL
arthropod	spruce beetles	CPL, CSEAL
female	Kate Mara	CPL, CMC
sport	BMX bicycling	CSEAL, CMC
profession	legal assistants	CPL
magazine	Thrasher	CPL
bird	Buff-throated Warbler	CSEAL
river	Fording River	CPL, CMC
mediaType	chemistry books	CPL, CMC
cityInState	(troy, Michigan)	CSEAL
musicArtistGenre	(Nirvana, Grunge)	CPL
tvStationInCity	(WLS-TV, Chicago)	CPL, CSEAL
sportUsesEquip	(soccer, balls)	CPL
athleteInLeague	(Dan Fouts, NFL)	RL
starredIn	(Will Smith, Seven Pounds)	CPL
productType	(Acrobat Reader, FILE)	CPL
athletePlaysSport	(scott shields, baseball)	RL
cityInCountry	(Dublin Airport, Ireland)	CPL

Table 1: Example beliefs promoted by NELL.

Results

After running for 67 days, NELL completed 66 iterations of execution. 242,453 beliefs were promoted across all predicates, 95% of which were instances of categories and 5% of relations. Example beliefs from a variety of predicates, along with the source components that extracted them, are shown in Table 1.

Following an initial burst of almost 10,000 beliefs promoted during the first iteration, NELL continued to promote a few thousand more on every successive iteration, indicating strong potential to learn more if it were left to run for a longer time. Figure 2 shows different views of the promotion activity of NELL over time. The left-hand figure shows overall numbers of promotions for categories and relations in each iteration. Category instances are promoted fairly steadily, while relation instance promotions are spiky. This is mainly because the RL component only runs every 10 iterations, and is responsible for many of the relation promotions. The right-hand figures are stacked bar plots showing the proportion of predicates with various levels of promotion activity during different spans of iterations. These plots show that instances are promoted for many different categories and relations during the whole run of NELL.

To estimate the precision of beliefs promoted during various stages of execution, we considered three time periods: iterations 1–22, iterations 23–44, and iterations 45–66. For each of these time periods, we uniformly sampled 100 beliefs promoted during those periods and judged their correctness. The results are shown in Table 2. During the three periods, the promotion rates are very similar, with between 76,000 and 89,000 instances promoted. There is a downward trend in estimated precision, going from 90% to 71% to 57%. Taking a weighted average of these three estimates of precision based on numbers of promotions, the overall estimated precision across all 66 iterations is 74%.

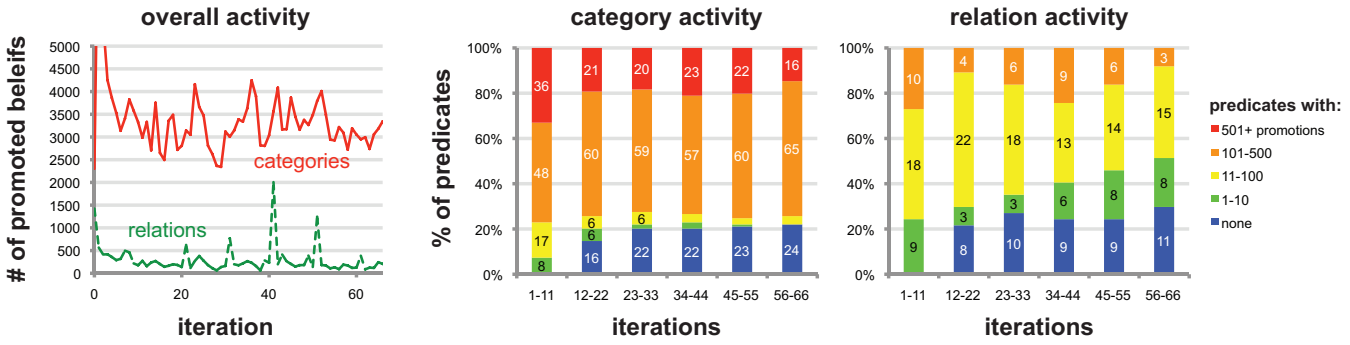

Figure 2: Promotion activity for beliefs over time. *Left:* The number of beliefs promoted for all category and relation predicates in each iteration. Periodic spikes among relation predicates occur every 10 iterations after the RL component runs. *Center and Right:* Stacked bar plots detailing the proportion of predicates (and counts of predicates, shown inside the bars) at various levels of promotion activity over time for categories and relations. Note that, while some predicates become "dormant" early on, the majority continue to show healthy levels of promotion activity even in later iterations of learning.

Iterations	Estimated Precision (%)	# Promotions
1–22	90	88,502
23–44	71	77,835
45–66	57	76,116

Table 2: Estimates of precision (from 100 sampled beliefs) and numbers of promoted beliefs across all predicates during iterations 1–22, 23–44, and 45–66. Note that the estimates of precision only consider beliefs promoted during a time period and ignore beliefs promoted earlier.

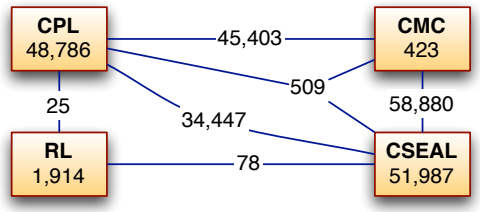

Figure 3: Source counts for beliefs promoted by NELL after 66 iterations. Numbers inside nodes indicate the number of beliefs promoted based solely on that component. Numbers on edges indicate beliefs promoted based on evidence from multiple components.

Only a few items were debated by the judges: examples are "right posterior," which was judged to not refer to a body part, and "green leafy salad," which was judged acceptable as a type of vegetable. "Proceedings" was promoted as a publication, which we considered incorrect (it was most likely due to noun-phrase segmentation errors within CPL). Two errors were due to languages ("Klingon Language" and "Mandarin Chinese language") being promoted as ethnic groups. ("Southwest", "San Diego") was labeled as an incorrect instance of the hasOfficesIn relation, since Southwest Airlines does not have an official corporate office there. Many system errors were subtle; one might expect a non-native reader of English to make similar mistakes.

To estimate precision at the predicate level, we randomly chose 7 categories and 7 relations which had at least 10 promoted instances. For each chosen predicate, we sampled 25 beliefs from iterations 1–22, 23–44, and 45–66, and judged their correctness. Table 3 shows these predicates and, for each time period, the estimates of precision and the number of beliefs promoted. Most predicates are very accurate, with precision exceeding 90%. Two predicates in particular, cardGame and productType, fare much worse. The cardGame category seems to suffer from the abundance of web spam related to casino and card games, which results in parsing errors and other problems. As a result of this noise, NELL ends up extracting strings of adjectives and nouns like "deposit casino bonuses free online list" as incorrect instances of cardGame. Most errors for the productType relation came from associating product names with more general nouns that are somehow related to the product but do not correctly indicate what kind of thing the product is, e.g., ("Microsoft Office", "PC"). Some of these productType beliefs were debated by the judges, but were ultimately labeled incorrect, e.g., ("Photoshop", "graphics"). In our ontology, the category for the second argument of productType is a general "item" super-category in the hierarchy; we posit that a more specific "product type" category might lead to more restrictive type checking.

As described in the Implementation section, NELL uses a Knowledge Integrator which promotes high-confidence single-source candidate facts, as well as candidate facts with multiple lower-confidence sources. Figure 3 illustrates the impact of each component within this integration strategy. Each component is shown containing a count which is the number of beliefs that were promoted based on that source alone having high confidence in that belief. Lines connecting components are labeled with counts that are the number of beliefs promoted based on those components each having some degree of confidence in that candidate. CPL and CSEAL each were responsible for many promoted beliefs on their own. However, more than half of the beliefs promoted by KI were based on multiple sources of evidence.

Predicate	Estimated Precision			# Promotions		
	1–22	23–44	45–66	1–22	23–44	45–66
cardGame	40	20	0	584	552	2,472
city	92	80	96	4,311	3,362	1,002
magazine	96	68	80	1,235	788	664
recordLabel	100	100	100	1,384	890	748
restaurant	96	88	92	242	568	523
scientist	96	100	100	768	1	404
vertebrate	100	100	96	1,196	1,362	714
athletePlaysForTeam	100	100	100	113	304	39
ceoOfCompany	100	100	100	82	8	9
coachesTeam	100	100	100	196	121	12
productType	28	44	20	35	156	195
teamPlaysAgainstTeam	96	100	100	283	553	232
teamPlaysSport	100	100	86	79	158	14
teamWonTrophy	88	72	44	119	104	174

Table 3: For selected categories (top) and relations (bottom), estimates of precision (from 25 sampled beliefs) and counts for beliefs promoted during iterations 1–22, 23–44, and 45–66.

Predicate	Pattern
emotion	hearts full of X
beverage	cup of aromatic X
newspaper	op-ed page of X
teamPlaysInLeague	X ranks second in Y
bookAuthor	Y classic X

Table 4: Example free-text patterns learned by CPL. X and Y represent placeholders for noun phrases to be extracted.

While RL was not responsible for many promoted beliefs, those that it did propose with high confidence appear to be largely independent from those of the other components.

RL learned an average of 66.5 novel rules per iteration, of which 92% were approved. 12% of the approved rules implied at least one candidate instance that had not yet been implied by another rule, and those rules implied an average of 69.5 such instances.

To give a sense of what is being learned by the different components used in NELL, we provide examples for each component. Table 4 shows contextual patterns learned by CPL. Table 5 shows web page wrappers learned by CSEAL. Example weights from the logistic regression classifiers learned by CMC are shown in Table 6. Finally, example rules induced by RL are shown in Table 7.

Supplementary Online Materials Several types of supplementary materials from our evaluation are posted online[3], including: (1) all promoted instances, (2) all categories, relations, and seed instances, (3) all labeled instances sampled for estimating precision, (4) all patterns promoted by CPL, and (5) all rules learned by RL.

[3] http://rtw.ml.cmu.edu/aaai10_online

Predicate	Feature	Weight
mountain	LAST=peak	1.791
mountain	LAST=mountain	1.093
mountain	FIRST=mountain	-0.875
musicArtist	LAST=band	1.853
musicArtist	POS=DT_NNS	1.412
musicArtist	POS=DT_JJ_NN	-0.807
newspaper	LAST=sun	1.330
newspaper	LAST=press	1.276
newspaper	LAST=university	-0.318
university	LAST=college	2.076
university	PREFIX=uc	1.999
university	LAST=university	1.745
university	FIRST=college	-1.381
visualArtMovement	SUFFIX=ism	1.282
visualArtMovement	PREFIX=journ	-0.234
visualArtMovement	PREFIX=budd	-0.253

Table 6: Example feature weights induced by the morphology classifier. Positive and negative weights indicate positive and negative impacts on predicted probabilities, respectively. Note that "mountain" and "college" have different weights when they begin or end an instance. The learned model uses part-of-speech features to identify typical music group names (e.g., The Beatles, The Ramones), as well as prefixes to disambiguate art movements from, say, academic fields and religions.

Discussion

These results are promising. NELL maintained high precision for many iterations of learning with a consistent rate of knowledge accumulation, all with a very limited amount of human guidance. We consider this to be significant progress toward our goal of building a never-ending language learner. In total, NELL learned 531 coupled functions, since 3 different subsystems (CMC, CPL, and CSEAL) learn about 123 categories, and 3 different subsystems (CPL, CSEAL, and RL) learn about 55 relations.

Predicate	Web URL	Extraction Template
academicField	http://scholendow.ais.msu.edu/student/ScholSearch.Asp	` [X] -`
athlete	http://www.quotes-search.com/d_occupation.aspx?o=+athlete	`<a href='d_author.aspx?a=[X]'>-`
bird	http://www.michaelforsberg.com/stock.html	`<option>[X]</option>`
bookAuthor	http://lifebehindthecurve.com/	`</li> <li>[X] by [Y] &#8211;`

Table 5: Examples of web page extraction templates learned by the CSEAL subsystem. $[X]$ and $[Y]$ represent placeholders for instances to be extracted (categories have only one placeholder; relations have two).

Probability	Consequent	Antecedents
0.95	athletePlaysSport(X, basketball)	$\Leftarrow$ athleteInLeague(X, NBA)
0.91	teamPlaysInLeague(X, NHL)	$\Leftarrow$ teamWonTrophy(X, Stanley Cup)
0.90	athleteInLeague(X, Y)	$\Leftarrow$ athletePlaysForTeam(X, Z), teamPlaysInLeague(Z, Y)
0.88	cityInState(X, Y)	$\Leftarrow$ cityCapitalOfState(X, Y), cityInCountry(X, USA)
† 0.62	newspaperInCity(X, New York)	$\Leftarrow$ companyEconomicSector(X, media), generalizations(X, blog)

Table 7: Example horn clauses induced by the rule learner. Probabilities indicate the conditional probability that the literal to the left of $\Leftarrow$ is true given that the literals to the right are satisfied. Each rule captures an empirical regularity among the relations mentioned by the rule. The rule marked with † was rejected during human inspection.

The stated goal for the system is to each day read more of the web to further populate its KB, and to each day learn to read more facts more accurately. As the KB growth over the past 67 days illustrate, the system does read more beliefs each day. Each day it also learns new extraction rules to further populate its KB, new extractors based on morphological features, new Horn clause rules that infer unread beliefs from other beliefs in the KB, and new URL-specific extractors that leverage HTML structure. Although NELL's ongoing learning allows it to extract more facts each day, the precision of the extracted facts declines slowly over time. In part this is due to the fact that the easiest extractions occur during early iterations, and later iterations demand more accurate extractors to achieve the same level of precision. However, it is also the case that NELL makes mistakes that lead to learning to make additional mistakes. Although we consider the current system promising, much research remains to be done.

The importance of our design principle of using components which make mostly independent errors is generally supported by the results. More than half of the beliefs were promoted based on evidence from multiple sources. However, in looking at errors made by the system, it is clear that CPL and CMC are not perfectly uncorrelated in their errors. As an example, for the category bakedGood, CPL learns the pattern "X are enabled in" because of the believed instance "cookies." This leads CPL to extract "persistent cookies" as a candidate bakedGood. CMC outputs high probability for phrases that end in "cookies," and so "persistent cookies" is promoted as a believed instance of bakedGood.

This behavior, as well as the slow but steady decline in precision of beliefs promoted by NELL, suggests an opportunity for leveraging more human interaction in the learning process. Currently, such interaction is limited to approving or rejecting inference rules proposed by RL. However, we plan to explore other forms of human supervision, limited to approximately 10–15 minutes per day. In particular, *active learning* (Settles 2009) holds much promise by allowing NELL to ask "queries" about its beliefs, theories, or even features about which it is uncertain. For example, a pattern like "X are enabled in" is only likely to occur with a few instances of the bakedGood category. This could be a poor pattern that leads to semantic drift, or it could be an opportunity to discover some uncovered subset of the bakedGood category. If NELL can adequately identify such opportunities for knowledge, a human can easily provide a label for this single pattern and convey a substantial amount of information in just seconds. Previous work has shown that labeling features (e.g., context patterns) rather than instances can lead to significant improvements in terms of reducing human annotation time (Druck, Settles, and McCallum 2009).

Conclusion

We have proposed an architecture for a never-ending language learning agent, and described a partial implementation of that architecture which uses four subsystem components that learn to extract knowledge in complimentary ways. After running for 67 days, this implementation populated a knowledge base with over 242,000 facts with an estimated precision of 74%.

These results illustrate the benefits of using a diverse set of knowledge extraction methods which are amenable to learning, and a knowledge base which allows the storage of candidate facts as well as confident beliefs. There are many opportunities for improvement, though, including: (1) self-reflection to decide what to do next, (2) more effective use of 10–15 minutes of daily human interaction, (3) discovery of new predicates to learn, (4) learning additional types of knowledge about language, (5) entity-level (rather than string-level) modeling, and (6) more sophisticated probabilistic modeling throughout the implementation.

Acknowledgments

This work is supported in part by DARPA (under contract numbers FA8750-08-1-0009 and AF8750-09-C-0179), Google, a Yahoo! Fellowship to Andrew Carlson, and the Brazilian research agency CNPq. We also gratefully acknowledge Jamie Callan for the ClueWeb09 web crawl and Yahoo! for use of their M45 computing cluster. We thank the anonymous reviewers for their helpful comments.

References

Anderson, J. R.; Byrne, M. D.; Douglass, S.; Lebiere, C.; and Qin, Y. 2004. An integrated theory of the mind. *Psychological Review* 111(4):1036–1050.

Banko, M., and Etzioni, O. 2007. Strategies for lifelong knowledge extraction from the web. In *Proc. of K-CAP*.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proc. of COLT*.

Callan, J., and Hoy, M. 2009. Clueweb09 data set. http://boston.lti.cs.cmu.edu/Data/clueweb09/.

Carbonell, J.; Etzioni, O.; Gil, Y.; Joseph, R.; Knoblock, C.; Minton, S.; and Veloso, M. 1991. PRODIGY: an integrated architecture for planning and learning. *SIGART Bull.* 2(4):51–55.

Carlson, A.; Betteridge, J.; Wang, R. C.; Jr., E. R. H.; and Mitchell, T. M. 2010. Coupled semi-supervised learning for information extraction. In *Proc. of WSDM*.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28:41–75.

Chang, M.-W.; Ratinov, L.-A.; and Roth, D. 2007. Guiding semi-supervision with constraint-driven learning. In *Proc. of ACL*.

Collins, M., and Singer, Y. 1999. Unsupervised models for named entity classification. In *Proc. of EMNLP*.

Curran, J. R.; Murphy, T.; and Scholz, B. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proc. of PACLING*.

Downey, D.; Etzioni, O.; and Soderland, S. 2005. A probabilistic model of redundancy in information extraction. In *Proc. of IJCAI*.

Druck, G.; Settles, B.; and McCallum, A. 2009. Active learning by labeling features. In *Proc. of EMNLP*.

Erman, L.; Hayes-Roth, F.; Lesser, V.; and Reddy, D. 1980. The HEARSAY-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys* 12(2):213–253.

Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2004. Methods for domain-independent information extraction from the web: an experimental comparison. In *Proc. of AAAI*.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*.

Laird, J.; Newell, A.; and Rosenbloom, P. 1987. SOAR: An architecture for general intelligence. *Artif. Intel.* 33:1–64.

Langley, P.; McKusick, K. B.; Allen, J. A.; Iba, W. F.; and Thompson, K. 1991. A design for the ICARUS architecture. *SIGART Bull.* 2(4):104–109.

Lenat, D. B. 1983. Eurisko: A program that learns new heuristics and domain concepts. *Artif. Intel.* 21(1-2):61–98.

Mitchell, T. M.; Allen, J.; Chalasani, P.; Cheng, J.; Etzioni, O.; Ringuette, M. N.; and Schlimmer, J. C. 1991. Theo: A framework for self-improving systems. *Arch. for Intelligence* 323–356.

Nahm, U. Y., and Mooney, R. J. 2000. A mutually beneficial integration of data mining and information extraction. In *Proc. of AAAI*.

Paşca, M.; Lin, D.; Bigham, J.; Lifchits, A.; and Jain, A. 2006. Names and similarities on the web: fact extraction in the fast lane. In *Proc. of ACL*.

Pennacchiotti, M., and Pantel, P. 2009. Entity extraction via ensemble semantics. In *Proc. of EMNLP*.

Quinlan, J. R., and Cameron-Jones, R. M. 1993. Foil: A midterm report. In *Proc. of ECML*.

Riloff, E., and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proc. of AAAI*.

Settles, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Thrun, S., and Mitchell, T. 1995. Lifelong robot learning. In *Robotics and Autonomous Systems*, volume 15, 25–46.

Wang, R. C., and Cohen, W. W. 2009. Character-level analysis of semi-structured documents for set expansion. In *Proc. of EMNLP*.

Yang, X.; Kim, S.; and Xing, E. 2009. Heterogeneous multitask learning with joint sparsity constraints. In *NIPS 2009*.

Yangarber, R. 2003. Counter-training in discovery of semantic patterns. In *Proc. of ACL*.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of ACL*.