Never-Ending Learning

T. Mitchell[*], W. Cohen[*], E. Hruschka[†¶], P. Talukdar[‡¶], J. Betteridge[*], A. Carlson[§¶], B. Dalvi[*], M. Gardner[*], B. Kisiel[*], J. Krishnamurthy[*], N. Lao[§¶], K. Mazaitis[*], T. Mohammad[¶], N. Nakashole[*], E. Platanios[*], A. Ritter[‖¶], M. Samadi[*], B. Settles[¶], R. Wang[¶], D. Wijaya[*], A. Gupta[*], X. Chen[*], A. Saparov[*], M. Greaves[††¶], J. Welling[‡‡]**

tom.mitchell@cs.cmu.edu

Abstract

Whereas people learn many different types of knowledge from diverse experiences over many years, most current machine learning systems acquire just a single function or data model from just a single data set. We propose a *never-ending learning* paradigm for machine learning, to better reflect the more ambitious and encompassing type of learning performed by humans. As a case study, we describe the Never-Ending Language Learner (NELL), which achieves some of the desired properties of a never-ending learner, and we discuss lessons learned. NELL has been learning to read the web 24 hours/day since January 2010, and so far has acquired a knowledge base with over 80 million confidence-weighted beliefs (e.g., *servedWith(tea, biscuits)*). NELL has also learned millions of features and parameters that enable it to read these beliefs from the web. Additionally, it has learned to reason over these beliefs to infer new beliefs, and is able to extend its ontology by synthesizing new relational predicates. NELL can be tracked online at http://rtw.ml.cmu.edu, and followed on Twitter at @CMUNELL.

Introduction

Machine learning is a highly successful branch of AI, and machine learning software is now widely used for tasks from spam filtering, to speech recognition, to credit card fraud detection, to face recognition. Despite this success, the ways in which computers learn today remain surprisingly narrow when compared to human learning. This paper explores an alternative paradigm for machine learning that more closely models the diversity, competence and cumulative nature of human learning. We call this alternative paradigm *never-ending learning*.

To illustrate, note that in each of the above examples the computer learns only a single function to perform a single task in isolation, usually from human labeled training examples of inputs and outputs of that function. In spam filtering, for instance, training examples consist of specific emails and spam or not-spam labels for each. This style of learning is often called *supervised function approximation*, because the abstract learning problem is to approximate some unknown function $f : X \rightarrow Y$ (e.g., the spam filter) given a training set of input/output pairs $\{\langle x_i, y_i \rangle\}$ of that function. Other machine learning paradigms exist as well (e.g., unsupervised clustering, topic modeling) but these paradigms also typically acquire only a single function or data model from a single dataset.

In contrast to these paradigms for learning single functions from well organized data sets over short time-frames, humans learn many different functions (i.e., different types of knowledge) over years of accumulated diverse experience, using extensive background knowledge learned from earlier experiences to guide subsequent learning.

The thesis of this paper is that *we will never truly understand machine or human learning until we can build computer programs that, like people,*

- *learn many different types of knowledge or functions,*
- *from years of diverse, mostly self-supervised experience,*
- *in a staged curricular fashion, where previously learned knowledge enables learning further types of knowledge,*
- *where self-reflection and the ability to formulate new representations and new learning tasks enable the learner to avoid stagnation and performance plateaus.*

We refer to this learning paradigm as "never-ending learning." The contributions of this paper are to (1) define more precisely the never-ending learning paradigm, (2) present as a case study a computer program called the Never-Ending Language Learner (NELL) which implements several of these capabilities, and which has been learning to read the web 24 hours/day for over four years, and (3) identify from NELL's strengths and weaknesses a number of key design features important to any never-ending learning system.

Related Work

Previous research has considered the problem of designing machine learning agents that persist over long periods of time (e.g., life long learning (Thrun and Mitchell 1995)), and that learn to learn (Thrun and Pratt 1998), yet there remain few if any working systems that demonstrate

[*]Carnegie Mellon University, USA
[†]University of São Carlos, Brazil
[‡]Indian Institute of Science, India
[§]Google Inc., USA
[¶]Research carried out while at Carnegie Mellon University
[‖]Ohio State University, USA
[**]Duolingo, USA
[††]Alpine Data Labs, USA
[‡‡]Pittsburgh Supercomputing Center, USA

this style of learning in practice. General architectures for problem solving and learning (e.g., SOAR (Laird, Newell, and Rosenbloom 1987), ICARUS (Langley et al. 1991), PRODIGY (Donmez and Carbonell 2008), THEO (Mitchell et al. 1991)) have been applied to problems from many domains, but again none of these programs has been allowed to learn continuously for any sustained period of time. Lenat's work on AM and Eurisko (Lenat 1983) represents an attempt to build a system that invents concepts, then uses these as primitives for inventing more complex concepts, but again this system was never allowed to run for a sustained period, because the author determined it would quickly reach a plateau in its performance.

Beyond such work on integrated agent architectures, there has also been much research on individual subproblems crucial to never-ending learning. For example, work on multitask transfer learning (Caruana 1997) suggests mechanisms by which learning of one type of knowledge can guide learning of another type. Work on active and proactive learning (Tong and Koller 2001; Donmez and Carbonell 2008) and on exploitation/exploration tradeoffs (Brunskill et al. 2012) presents strategies by which learning agents can collect optimal training data from their environment. Work on learning of latent representations (Bengio 2009; Muggleton and Buntine 1992) provides methods that might enable never-ending learners to expand their internal knowledge representations over time, thereby avoiding plateaus in performance due to lack of adequate representations. Work on curriculum learning (Bengio et al. 2009) explores potential synergies across sets or sequences of learning tasks. Theoretical characterizations of cotraining (Blum and Mitchell 1998) and other multitask learning methods (Balcan and Blum 2004; Platanios, Blum, and Mitchell 2014) have provided insights into when and how the sample complexity of learning problems can be improved via multitask learning.

Despite this relevant previous research, we remain in the very early stages in studying never-ending learning methods. We have almost no working systems to point to, and little understanding of how to architect a computer system that successfully learns over a prolonged period of time, while avoiding plateaus in learning due to saturation of learned knowledge. The key contributions of this paper are first, to present a working case study system, an extended version of an early prototype reported in (Carlson et al. 2010a), which successfully integrates a number of key competencies; second, an empirical evaluation of the prototype's performance over time; and third, an analysis of the prototype's key design features and shortcomings, relative to the goal of understanding never-ending learning.

Never-Ending Learning

Informally, we define a never-ending learning agent to be a system that, like humans, learns many types of knowledge, from years of diverse and primarily self-supervised experience, using previously learned knowledge to improve subsequent learning, with sufficient self-reflection to avoid plateaus in performance as it learns. The never-ending learning *problem* faced by the agent consists of a collection of learning tasks, and constraints that couple their solutions.

To be precise, we define a *never-ending learning problem* $\mathscr{L}$ to be an ordered pair consisting of: (1) a set $L = \{L_i\}$ of learning tasks, where the ith *learning task* $L_i = \langle T_i, P_i, E_i \rangle$ is to improve the agent's performance, as measured by metric P_i, on a given *performance task* T_i, through a given type of *experience* E_i; and (2) a set of coupling constraints $C = \{\langle \phi_k, V_k \rangle\}$ among the solutions to these learning tasks, where ϕ_k is a real-valued function over two or more learning tasks, specifying the degree of satisfaction of the constraint, and V_k is a vector of indices over learning tasks, specifying the arguments to ϕ_k.

$$\begin{aligned} \mathscr{L} &= (L, C) \\ \text{where, } L &= \{\langle T_i, P_i, E_i \rangle\} \\ C &= \{\langle \phi_k, V_k \rangle\} \end{aligned} \tag{1}$$

Above, each performance task T_i is a pair $T_i \equiv \langle X_i, Y_i \rangle$ defining the domain and range of a function to be learned $f_i^* : X_i \rightarrow Y_i$. The performance metric $P_i : f \rightarrow \mathbb{R}$ defines the optimal learned function f_i^* for the ith learning task:

$$f_i^* \equiv \arg\max_{f \in F_i} P_i(f)$$

where F_i is the set of all possible functions from X_i to Y_i.

Given such a learning *problem* containing n learning tasks, a never-ending learning *agent* $\mathscr{A}$ outputs a sequence of solutions to these learning tasks. As time passes, the quality of these n learned functions should improve, as measured by the individual performance metrics $P_1 \ldots P_n$ and the degree to which the coupling constraints C are satisfied.

To illustrate, consider a mobile robot with sensor inputs S and available actions A. One performance task, $\langle S, A \rangle$, might be for the robot to choose actions to perform from any given state, and the corresponding learning task $\langle \langle S, A \rangle, P_1, E_1 \rangle$ might be to learn the specific function $f_1 : S \rightarrow A$ that leads most quickly to a goal state defined by performance metric P_1, from training experience E_1 obtained via human teleoperation. A second performance task for the same robot may be to predict the outcome of any given action in any given state: $\langle S \times A, S \rangle$. Here, the learning task $\langle \langle S \times A, S \rangle, P_2, E_2 \rangle$ might be to learn this prediction function $f_2 : S \times A \rightarrow S$ with *high accuracy* as specified by performance metric P_2, from experience E_2 consisting of the robot wandering autonomously through its environment.

Note these two robot learning tasks can be coupled by enforcing the constraint that the learned function f_1 must choose actions that do indeed lead optimally to the goal state according to the predictions of learned function f_2. By defining this coupling constraint $\phi(L_1, L_2)$ between the solutions to these two learning tasks, we give the learning agent a chance to improve its ability to learn one function by success in learning the other.

We are interested in never-ending Learning agents that address such never-ending learning problems $\mathscr{L} = (L, C)$, especially in which the learning agent

- *learns many different types of knowledge*; that is, L contains many learning tasks
- *from years of diverse, primarily self-supervised experience*; that is, the experiences $\{E_i\}$ on which learning is

based are realistically diverse, and largely provided by the system itself,

- *in a staged, curricular fashion where previously learned knowledge supports learning subsequent knowledge*; that is, the different learning tasks $\{L_i\}$ need not be solved simultaneously – solving one helps solve the next, and
- *where self-reflection and the ability to formulate new representations, new learning tasks, and new coupling constraints enables the learner to avoid becoming stuck in performance plateaus*; that is, where the learner may itself add new learning tasks and new coupling constraints that help it address the given learning problem $\mathscr{L}$.

Case Study: Never Ending Language Learner

The Never Ending Language Learner (NELL), an early prototype of which was reported in (Carlson et al. 2010a), is a learning agent whose task is to learn to read the web. The input-output specification of NELL's task is:

Given:

- an initial ontology defining categories (e.g., Sport, Athlete) and binary relations (e.g., AthletePlaysSport(x,y)),
- approximately a dozen labeled training examples for each category and relation (e.g., examples of Sport might include the noun phrases "baseball" and "soccer"),
- the web (an initial 500 million web pages from the ClueWeb 2009 collection (Callan and Hoy 2009), and access to 100,000 Google API search queries each day),
- occasional interaction with humans (e.g., through NELL's public website http://rtw.ml.cmu.edu);

Do: Run 24 hours/day, forever, and each day:

1. read (extract) more beliefs from the web, and remove old incorrect beliefs, to populate a growing knowledge base containing a confidence and provenance for each belief,
2. learn to read better than the previous day.

NELL has been running since January 2010, each day extracting more beliefs from the web, then retraining itself to improve its competence. The result so far is a knowledge base (KB) with over 80 million interconnected beliefs (see Figure 1), along with millions of learned phrasings, morphological features, and web page structures NELL now uses to extract beliefs from the web. NELL is also now learning to reason over its extracted knowledge to infer new beliefs it has not yet read, and it is now able to propose extensions to its initial manually-provided ontology.

NELL's Never Ending Learning Problem

Above we described the input-output specification of the NELL system. Here we describe NELL's never-ending learning problem $\langle L, C \rangle$ in terms of the general formalism introduced in section 2, first describing NELL's learning tasks L, then its coupling constraints C. The subsequent section describes NELL's approach to this never-ending learning problem, including NELL's mechanisms for adding its own new learning tasks and coupling constraints.

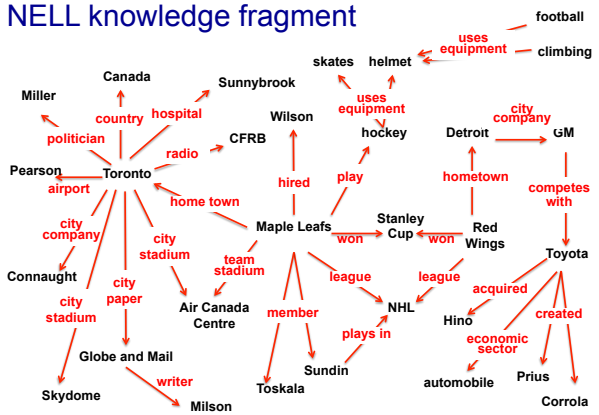

Figure 1: **Fragment of the 80 million beliefs NELL has read from the web.** Each edge represents a belief triple (e.g., play(MapleLeafs, hockey), with an associated confidence and provenance not shown here. This figure contains only correct beliefs from NELL's KB – it has many incorrect beliefs as well since NELL is still learning.

NELL's Learning Tasks: Following the notation in Equation 1, each of NELL's learning tasks consists of a performance task, performance metric, and type of experience $\langle T_i, P_i, E_i \rangle$. NELL faces over 2500 distinct learning tasks, corresponding to distinct functions $f_i : X_i \rightarrow Y_i$ it is trying to learn for its distinct performance tasks $T_i = \langle X_i, Y_i \rangle$. These tasks fall into several broad groups:

- *Category Classification*: Functions that classify noun phrases by semantic category (e.g., a boolean valued function that classifies whether any given noun phrase refers to a food). NELL learns different boolean functions for each of the 280 categories in its ontology, allowing noun phrases to refer to entities in multiple semantic categories (e.g., "apple" can refer to a "Food" as well as a "Company"). For each category Y_i NELL learns up to five distinct functions that predict Y_i, based on five different views of the noun phrase (five different X_i's), which are:

  1. *Character string features* of the noun phrase (e.g., whether the noun phrase ends with the character string "...burgh"). This is performed by the CML system (Carlson et al. 2010b), which represents the noun phrase by a vector with thousands of string features.
  2. *The distribution of text contexts found around this noun phrase in 500M English web pages* from the ClueWeb2009 text corpus (Callan and Hoy 2009) (e.g., how frequently the noun phrase N occurs in the context "mayor of N"). This is performed by the CPL system (Carlson et al. 2010b).
  3. *The distribution of text contexts found around this noun phrase through active web search*. This is performed by the OpenEval system (Samadi, Veloso, and Blum 2013), which uses somewhat different context features from the above CPL system, and uses real time web search to collect this information.

4. *HTML structure of web pages containing the noun phrase* (e.g., whether the noun phrase appears in an HTML list, alongside other known cities). This is performed by the SEAL system (Wang and Cohen 2007).
5. *Visual images* associated with this noun phrase, when the noun phrase is given to an image search engine. This is performed by the NEIL system (Chen, Shrivastava, and Gupta 2013), and applies only to a subset of NELL's ontology categories (e.g., not to MusicGenre).

- *Relation Classification*: These functions classify pairs of noun phrases by whether or not they satisfy a given relation (e.g., classifying whether the pair ⟨"Pittsburgh","U.S."⟩ satisfies the relation "CityLocatedInCountry(x,y)"). NELL learns distinct boolean-valued classification functions for each of the 327 relations in its ontology. For each relation, NELL learns three distinct classification functions based on different feature views of the input noun phrase pair. Specifically, it uses the two classification methods CPL and OpenEval based on the distribution of text contexts found between the two noun phrases on web pages, and it uses the SEAL classification method based on HTML structure of web pages.
- *Entity Resolution*: Functions that classify noun phrase pairs by whether or not they are synonyms (e.g., whether "NYC" and "Big Apple" can refer to the same entity). This classification method is described in (Krishnamurthy and Mitchell 2011). For each of NELL's 280 categories, it co-trains two synonym classifiers: one based on string similarity between the two noun phrases, and a second based on similarities in their extracted beliefs.
- *Inference Rules among belief triples*: Functions that map from NELL's current KB, to new beliefs it should add to its KB. For each relation in NELL's ontology, the corresponding function is represented by a collection of restricted Horn Clause rules learned by the PRA system (Lao, Mitchell, and Cohen 2011; Gardner et al. 2014).

Each of the above functions $f : X \rightarrow Y$ represents a performance task $T_i = \langle X, Y \rangle$ for NELL, and each maps to the learning task of acquiring that function, given some type of experience E_i and a performance metric P_i to be optimized during learning. In NELL, the performance metric P_i to optimize is simply the *accuracy* of the learned function. In all cases except one, the experience E_i is a combination of *human-labeled training examples* (the dozen or so labeled examples provided for each category and relation in NELL's ontology, plus labeled examples contributed over time through NELL's website), a set of *NELL self-labeled training examples* corresponding to NELL's current knowledge base, and a huge volume of unlabeled web text. The one exception is learning over visual images, which is handled by the NEIL system with its own training procedures.

NELL's Coupling Constraints: The second component of NELL's never-ending learning task is the set of *coupling constraints* which link its learning tasks. NELL's coupling constraints fall into five groups:

- *Multi-view co-training coupling.* NELL's multiple methods for classifying noun phrases into categories (and noun phrase pairs into relations) provide a natural co-training setting (Blum and Mitchell 1998), in which alternative classifiers for the same category should agree on the predicted label whenever they are given the same input, even though their predictions are based on different noun phrase features. To be precise, let $v_k(z)$ be the feature vector used by the kth function, when considering input noun phrase z. For any pair of functions $f_i : v_i(Z) \rightarrow Y$ and $f_j : v_j(Z) \rightarrow Y$ that predict the same Y from the same Z using the two different feature views v_i and v_j, NELL uses the coupling constraint $(\forall z) f_i(z) = f_j(z)$. This couples the tasks of learning f_i and f_j.
- *Subset/superset coupling.* When a new category is added to NELL's ontology, the categories which are its immediate parents (supersets) are specified (e.g., "Beverage" is declared to be a subset of "Food."). When category $C1$ is added as a subset of category $C2$, NELL uses the coupling constraint that $(\forall x) C1(x) \rightarrow C2(x)$. This couples learning tasks that learn to predict $C1$ to those that learn to predict $C2$.
- *Multi-label mutual exclusion coupling.* When a category C is added to NELL's ontology, the categories that are known to be disjoint from (mutually exclusive with) C are specified (e.g., "Beverage" is declared to be mutually exclusive with "Emotion," "City", etc.). These mutual exclusion constraints are typically inherited from more general classes, but can be overridden by explicit assertions. When category $C1$ is declared to be mutually exclusive with $C2$, NELL adopts the constraint that $(\forall x) C1(x) \rightarrow \neg C2(x)$.
- *Coupling relations to their argument types.* When a relation is added to NELL's ontology, the editor must specify the types of its arguments (e.g., that "zooInCity(x,y)" requires arguments of types "Zoo" and "City" respectively). NELL uses these argument type declarations as coupling constraints between its category and relation classifiers.
- *Horn clause coupling.* Whenever NELL learns a Horn clause rule to infer new KB beliefs from existing beliefs, that rule serves as a coupling constraint to augment NELL's never ending learning problem $\langle L, C \rangle$. For example, when NELL learns a rule of the form $(\forall x, y, z) R_1(x, y) \wedge R_2(y, z) \rightarrow R_3(x, z)$ with probability p, this rule serves as a new probabilistic coupling constraint over the functions that learn relations R_1, R_2, and R_3. Each learned Horn clause requires that learned functions mapping from noun phrase pairs to relations labels for R_1, R_2, and R_3 are consistent with this Horn clause; hence, they are analogous to NELL's subset/superset coupling constraints, which require that functions mapping from noun phrases to category labels should be consistent with the subset/superset constraint.

NELL's never ending learning problem thus contains over 2500 learning tasks, inter-related by over a million coupling constraints. In fact, NELL's never ending learning problem $\langle L, C \rangle$ is open ended, in that NELL has the ability to add both new consistency constraints in the form of learned Horn clauses (as discussed above) and new learning tasks, by in-

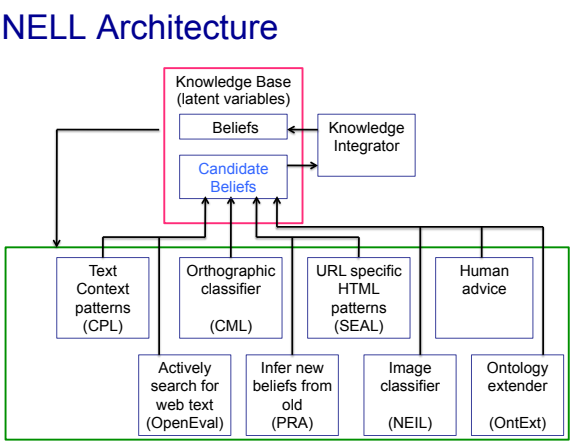

Figure 2: **NELL's software architecture.** NELL's growing knowledge base (KB) serves as a shared blackboard through which its various reading and inference modules interact. NELL's learning cycle iteratively retrains these software modules using the current KB, then updates the KB using these refined modules.

venting new predicates for its ontology (as discussed below).

NELL's Learning Methods and Architecture

The software architecture for NELL, depicted in Figure 2, includes a knowledge base (KB) which acts as a blackboard through which NELL's various learning and inference modules communicate.[1] As shown in the figure, these software modules map closely to the learning methods (CPL, CML, SEAL, OpenEval, PRA, NEIL) for the different types of functions mentioned in the previous section, so that NELL's various learning tasks are partitioned across these modules.

Learning in NELL as an Approximation To EM: NELL is in an infinite loop analogous to an EM algorithm for semi-supervised learning, performing an E-like step and an M-like step on each iteration. During the E-like step, each reading and inference module proposes updates to the KB (additions and deletions of specific beliefs, with specific confidences and provenance information). The Knowledge Integrator (KI) both records these individual recommendations and makes a final decision about the confidence assigned to each potential belief in the KB. Then, during the M-like step, this refined KB is used to retrain each of these software modules, employing module-specific learning algorithms. The result is a large-scale coupled training system in which thousands of learning tasks are guided by one another's results, through the shared KB and coupling constraints.

Notice that a full EM algorithm is impractical in NELL's case; NELL routinely considers tens of millions of noun phrases, yielding 10^{17} potential relational assertions among noun phrase pairs. It is impractical to estimate the probability of each of these potential latent assertions on each E-like step. Instead, NELL constructs and considers only the beliefs in which it has highest confidence, limiting each software module to suggest only a bounded number of new candidate beliefs for any given predicate on any given iteration. This enables NELL to operate tractably, while retaining the ability to add millions of new beliefs over many iterations.

Knowledge Integrator in NELL: The Knowledge Integrator (KI) integrates the incoming proposals for KB updates. For efficiency, the KI considers only moderate-confidence candidate beliefs, and re-assesses confidence using a limited subgraph of the full graph of consistency constraints and beliefs. As an example, the KI considers all beliefs in the current KB to assure that argument types are satisfied for new relational assertions, but does not consider possible updates to beliefs about these argument types in the same iteration. Over multiple iterations, the effects of constraints propagate more widely through this graph of beliefs and constraints. Recently, (Pujara et al. 2013) has demonstrated a more effective algorithm for the joint inference problem faced by the KI; we are now in the process of upgrading NELL's KI to use this implementation.

Adding Learning Tasks and Ontology Extension in NELL: NELL has the ability to extend its ontology by inventing new relational predicates using the OntExt system (Mohamed, Hruschka Jr., and Mitchell 2011). OntExt considers every pair of categories in NELL's current ontology, to search for evidence of a frequently discussed relation between members of the category pair, in a three step process: (1) Extract sentences mentioning known instances of both categories (e.g., for the category pair $\langle drug, disease \rangle$ the sentence *Prozac may cause migraines* might be extracted if *prozac* and *migraines* were already present in NELL's KB). (2) From the extracted sentences, build a context by context co-occurrence matrix, then cluster the related contexts together. Each cluster corresponds to a possible new relation between the two input category instances. (3) Employ a trained classifier, and a final stage of manual filtering, before allowing the new relation (e.g., DrugHasSideEffect(x,y)) to be added to NELL's ontology. OntExt has added 62 new relations to NELL's ontology. Note each new relation spawns associated new learning tasks, including three new tasks of learning to classify which noun phrase pairs satisfy the relation (based on different views of the noun phrase pair), and a task of learning Horn clause rules to infer this new relation from others.

Empirical Evaluation

Our primary goal in experimentally evaluating NELL is to understand the degree to which NELL improves over time through learning, both in its reading competence, and in the size and quality of its KB.

First, consider the growth of NELL's KB over time, from its inception in January 2010 through November 2014, during which NELL has completed 886 iterations. The left panel of Figure 3 shows the number of beliefs in NELL's KB

[1] The KB is implemented as a frame-based knowledge representation which represents language tokens (e.g., NounPhrase:bank) distinct from non-linguistic entities to which they can refer (e.g., Company:bank, LandscapeFeature:bank), and relates the two by separate CanReferTo(noun phrase, entity) assertions.

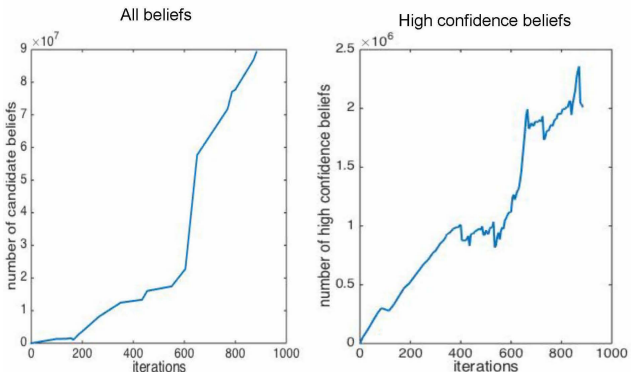

Figure 3: **NELL KB size over time.** Total number of beliefs (left) and number of high confidence beliefs (right) versus iterations. Left plot vertical axis is tens of millions, right plot vertical axis is in millions.

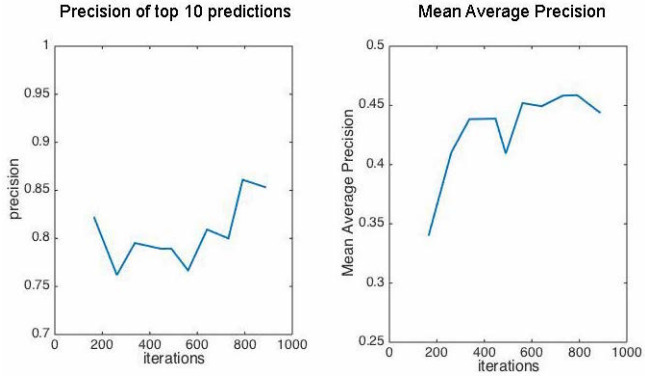

Figure 4: **NELL reading accuracy over time.** Left plot shows accuracy of 310 novel predictions, consisting of the 10 most confident predictions for each of 31 predicates, omitting any that correspond to human labeled examples. Right plot shows the Mean Average Precision over 1000 most confident predictions for the same 31 predicates.

over time, and the right panel of this figure shows the number of beliefs for which NELL holds high confidence. Note in November 2014, NELL has approximately 89 million beliefs with varying levels of confidence, 2 million of which it holds in high confidence. Here, "high confidence" indicates either that one of NELL's modules assigns a confidence of at least 0.9 to the belief, or that multiple modules independently propose the belief. NELL's KB is clearly growing, though its high confidence beliefs are growing more slowly than its total set of beliefs. Note also the growth in high confidence beliefs has diminished somewhat in the most recent iterations. This is in part due to the fact that NELL has saturated some of the categories and relations in its ontology. For example, for the category "Country" it extracted most actual country names in the first few hundred iterations.

Second, consider the accuracy of NELL's reading competence over time. To evaluate this, we applied different versions of NELL obtained at different iterations in its history, to extract beliefs from a fixed set of text data consisting of the 500 million English web pages from the ClueWeb2009 corpus, plus the world wide web as of November 14, 2014. We then manually evaluated the accuracy of the beliefs extracted by these different historical versions of NELL, to measure NELL's evolving reading competence. To obtain different versions of NELL over time, we relied on the fact that NELL's state at any given time is fully determined by its KB. In particular, given NELL's KB at iteration i we first had NELL train itself on that KB plus unlabeled text, then had it apply its trained methods to a fixed set of unlabeled web text to propose a rank-ordered set of confidence-weighted beliefs. We evaluated the accuracy of these beliefs to measure NELL's evolving competence at different points in time.

In greater detail, we first selected 10 different points in time to test NELL's KB: iterations 166, 261, 337, 447, 490, 561, 641, 731, 791, and 886. For each of those iterations, we trained NELL using the KB from that iteration, then evaluated its reading competence over a representative sample of 18 categories and 13 relations (31 predicates in total) from NELL's current ontology of 280 categories and 327 relations. Each iteration-specific version of NELL's readers was applied to produce a ranked list of the top 1000 novel predictions for each predicate over all of ClueWeb (i.e., ignoring any instance for which NELL has received human input). We then created a pool of instances to manually annotate. For each iteration, we included the top 10 ranked predictions for each predicate, 20 more predications sampled uniformly at random from ranks 11–100, and an additional 20 from ranks 101–1000. This provided 50 (potentially overlapping) instances per predicate from each iteration, averaging about 350 instances per predicate over all iterations. We manually annotated each of these instances as correct or incorrect, yielding approximately 11,000 total annotated beliefs regarding 31 predicates, which we used to evaluate NELL at each iteration.

First consider the mean precision of NELL's 10 highest ranked novel predictions for each of the 31 predicates (310 predictions in total), at each of these 10 points in time. As shown in Figure 4, this measure of NELL's precision shows a gradual upward trend over time, with NELL's recent iterations yielding the highest precision, currently at 0.85 (precision of the top 25 predictions is 0.84). Note that iteration 166 also has relatively high precision compared to iterations 261 through 731. This is likely to be the result of an abnormally high rate of human feedback to NELL during iterations 100 through 166, resulting in a temporary boost in accuracy. During these 67 iterations NELL received 31% of the total human feedback it has received over its 866 iterations. A second measure of NELL's accuracy , shown in the right panel of this figure, is the mean average precision (MAP) over the sample of data drawn from the top 1000 predications for the same 31 predicates. Here again, we see a gradual improvement in NELL's accuracy over its top 1000 novel predications per predicate, indicating that NELL is indeed improving its reading competence over time.

Next, we summarize feedback from humans to NELL, which over time has been dominated by negative feedback identifying NELL's incorrect beliefs. Over the 58 months

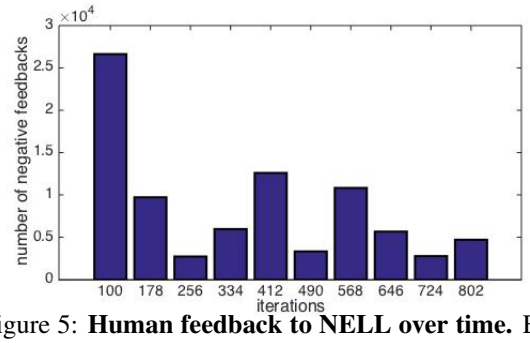

Figure 5: **Human feedback to NELL over time.** Each bar in this histogram shows the number of NELL beliefs for which humans provided negative feedback, during a 78 iteration interval. This averages out to 2.4 items of feedback per month, per predicate in NELL's ontology, over NELL's 58 months of operation.

of NELL's existence, it has averaged 2.4 negative feedbacks per predicate, per month, for a total of 85,088 items of negative feedback, or 1,467 per month. Figure 5 shows the distribution of this feedback over time. Note the large burst of feedback from iteration 100 to 177. During the first two years, the bulk of feedback was provided by members of the NELL research project, though in more recent years much of the feedback is now crowdsourced, i.e., provided by external visitors to the NELL website.

In addition to the above aggregate measures of NELL's behavior, it is interesting to consider its detailed behavior for specific predicates. Here we find that NELL's performance varies dramatically across predicates: the precision over NELL's 1000 highest confidence predictions for categories such as "river," "body part," and "physiological condition" is well above 0.95, whereas for "machine learning author" and "city capital of country(x,y)" accuracies are well below 0.5. One factor influencing NELL's ability to learn well is having other mutually exclusive categories to learn, which provide nearby negative examples. For instance, many of NELL's errors for the category "machine learning author" are computer science researchers (e.g., "Robert Kraut") who do not happen to work in the area of machine learning – NELL would presumably learn this category better if we added categories such as "HCI author" to provide examples that are usually mutually exclusive. Another factor is the number of actual members of the category: for example, the category "planet" has only a small number of actual members, but NELL is searching for more, so it proposes members such as "counter earth" and "asteroid ida." In some cases, NELL fails on a predicate due to a particular error which propagates due to its bootstrap learning. For example, for the category "sports team position" NELL has numerous correct members such as "quarterback" and "first base," but it has acquired a systematic error in having a strong belief that phrases ending with "layer" (e.g., "defence layer" and "cloud layer") refer to sports positions. While some do, most do not, yet NELL has no easy way to determine this.

Based on the above empirical analysis, it is clear that NELL is successfully learning to improve its reading competence over time, and is using this increasing competence to build an ever larger KB of beliefs about the world. Importantly, this evaluation also shows a slowing of NELL's KB growth rate, at least for high confidence beliefs. In part this is due to the fact that as NELL matures, the task of adding new knowledge to the KB naturally becomes more difficult: NELL's redundancy-based reading methods tend to extract the most frequently-mentioned beliefs earlier (e.g., emotions such as 'gladness' and 'loneliness') so that later it can only grow the KB by extracting less frequently mentioned beliefs which are more difficult to extract (e.g., emotions such as 'incredible lightness,' 'cavilingness,' and ' "nonopprobriousness'). Meeting this challenge of increasing difficulty over time in adding new KB beliefs suggests several opportunities for future research: (1) add a self-reflection capability to NELL to enable it to detect where it is doing well, where it is doing poorly, when it has sufficiently populated any given category or relation, enabling it to allocate its efforts in a more intelligently targeted fashion, (2) broaden the scope of data NELL uses to extract beliefs, for example by including languages beyond English, image data, and Twitter, (3) expand NELL's ontology dramatically, both by relying more heavily on automated algorithms for inventing new relations and categories, and by merging other open-source ontologies such as DBpedia into NELL's ontology (**?**), and (4) add a new generation of "micro-reading" methods to NELL – methods that perform deep semantic analysis of individual sentences and text passages, and which therefore do not need to rely on redundancy across the web to achieve accurate reading. We are currently actively exploring each of these directions.

Discussion

NELL is a learning agent that demonstrates some of the properties we believe will be important to any never-ending learning system, though it has limitations as well. Our experience with NELL suggests four useful design features that we recommend for any never-ending learning system:

- *To achieve successful semi-supervised learning, couple the training of many different learning tasks.* The primary reason NELL has succeeded in learning thousands of functions from only a small amount of supervision is that it has been designed to simultaneously learn thousands of different functions that are densely connected by a large number of coupling constraints. As progress begins to be made on one of these learning tasks, the coupling constraints allow the learned information to constrain subsequent learning for other tasks.
- *Allow the agent to learn additional coupling constraints.* Given the critical importance of coupling the training of many functions, great gains can be had by automatically learning additional coupling constraints. In NELL, this is accomplished by learning restricted-form probabilistic Horn clauses by data-mining NELL's KB. NELL has learned tens of thousands of probabilistic Horn clauses which it uses to infer new KB beliefs it has not yet read. As a side effect of creating new beliefs which are subsequently used to retrain NELL's reading functions, these Horn clauses also act as coupling constraints to further

constrain and guide subsequent learning of NELL's reading functions for relations mentioned by the Horn clause.

- *Learn new representations that cover relevant phenomena beyond the initial representation.* To continuously improve, and to avoid reaching a plateau in performance, a never-ending learning system may need to extend its representation beyond what is initially provided. NELL has a primitive but already-useful ability to extend its representation by suggesting new relational predicates (e.g., RiverFlowsThroughCity(x,y)) between existing categories (e.g., river, city). Each new relation NELL introduces leads to new learning tasks such as learning to extract the relation from text, and learning to infer instances of the relation from other beliefs.
- *Organize the set of learning tasks into an easy-to-increasingly-difficult curriculum.* Given a complex set of learning tasks, it will often be the case that some learning tasks are easier, and some produce pre-requisite knowledge for others. In NELL, we have evolved the system by manually introducing new types of learning tasks over time. During NELL's first six months, its only tasks were to classify noun phrases into categories, and noun phrase pairs into relations. Later, once it achieved some level of competence at these, and grew its KB accordingly, it became feasible for it to confront more challenging tasks. At that point, we introduced the task of datamining the KB to discover useful Horn clause rules, as well as the task of discovering new relational predicates based on NELL's knowledge of category instances. A key open research question is how the learning agent might itself evolve a useful curriculum of learning tasks.

NELL also has many limitations, which suggest additional areas for research into never-ending learning agents:

- *Self reflection and an explicit agenda of learning subgoals.* At present, NELL suffers from the fact that it has a very weak ability to monitor its own performance and progress. It does not notice, for example, that it has learned no useful new members of the "country" category for the past year, and it continues to work on this problem although its knowledge in this area is saturated. Furthermore, it makes no attempt to allocate its learning effort to tasks that will be especially productive (e.g., collecting new web text describing entities about which it has only low confidence beliefs). It is clear that developing a self-reflection capability to monitor and estimate its own accuracy, and to plan specific learning actions in response to perceived needs, would allow the system to use its computational effort more productively.
- *Pervasive plasticity.* Although NELL is able to modify many aspects of its behavior through learning, other parts of its behavior are cast in stone, unmodifiable. For example, NELL's method for detecting noun phrases in text is a fixed procedure not open to learning. In designing never-ending learning agents, it will be important to understand how to architect the agent so that as many aspects of its behavior as possible are plastic–i.e., open to learning. Otherwise, the agent runs the risk of reaching a performance plateau in which further improvement requires modifications to a part of the system that is not itself modifiable.
- *Representation and reasoning.* At present, NELL uses a simple frame based knowledge representation, augmented by the PRA reasoning system which performs tractable but limited types of reasoning based on restricted Horn clauses. NELL's competence is already limited in part by its lack of more powerful reasoning components: currently lacks methods for representing and reasoning about time and space. Hence, core AI problems of representation and tractable reasoning are also core research problems for never-ending learning agents.

The study of never-ending learning raises important conceptual and theoretical problems as well, including:

- *The relationship between* **consistency** *and* **correctness**. An autonomous learning agent can never truly perceive whether it is correct – it can at best detect only that it is internally consistent. For example, even if it observes that its predictions (e.g., new beliefs predicted by NELL's learned Horn clauses) are consistent with what it perceives (e.g., what NELL reads from text), it cannot distinguish whether that observed *consistency* is due to correct predictions, or incorrect perceptions. This is important in understanding never-ending learning, because it suggests organizing the learning agent to become increasingly consistent over time, which is precisely how NELL uses its consistency constraints to guide learning. A key open theoretical question therefore is "under what conditions can one guarantee that an increasingly consistent learning agent is also an increasingly correct agent?" (Platanios, Blum, and Mitchell 2014) provides one step in this direction, by providing an approach that will soon allow NELL to estimate its accuracy based on the observed consistency rate among its learned functions, but much remains to be understood about this fundamental theoretical question.
- *Convergence guarantees in principle and in practice.* A second fundamental question for never-ending learning agents is "what agent architecture is sufficient to guarantee that the agent can in principle generate a sequence of self-modifications that will transform it from its initial state to an increasingly high performance agent, without hitting performance plateaus?" Note this may require that the architecture support pervasive plasticity, the ability to change its representations, etc. One issue here is whether the architecture has sufficient self-modification operations to allow it to produce ever-improving modifications to itself *in principle*. A second, related issue is whether its learning mechanisms will make these potential changes, converging *in practice* given a tractable amount of computation and training experience.

Acknowledgment

We thank the anonymous reviewers for their constructive comments. This research was supported in part by DARPA under contract number FA8750-13-2-0005, NSF grants IIS-1065251 and CCF-1116892, and Google. We also gratefully

acknowledge fellowship support from Yahoo, Microsoft, Google, and Fulbright.

References

Balcan, M.-F., and Blum, A. 2004. A PAC-style model for learning from labeled and unlabeled data. In *Proc. of COLT*.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48. ACM.

Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning* 2(1):1–127.

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

Brunskill, E.; Leffler, B.; Li, L.; Littman, M. L.; and Roy, N. 2012. Corl: A continuous-state offset-dynamics reinforcement learner. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)*, 53–61.

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

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka Jr, E. R.; and Mitchell, T. M. 2010a. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, 3.

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

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

Chen, X.; Shrivastava, A.; and Gupta, A. 2013. Neil: Extracting visual knowledge from web data. In *In Proceedings of ICCV*.

Donmez, P., and Carbonell, J. G. 2008. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, 619–628. ACM.

Gardner, M.; Talukdar, P.; Krishnamurthy, J.; and Mitchell, T. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of EMNLP 2014*.

Krishnamurthy, J., and Mitchell, T. M. 2011. Which noun phrases denote which concepts. In *Proceedings of ACL 2011*.

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.

Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of EMNLP 2011*.

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.

Mohamed, T.; Hruschka Jr., E. R.; and Mitchell, T. M. 2011. Discovering relations between noun categories. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1447–1455. Edinburgh, Scotland, UK.: Association for Computational Linguistics.

Muggleton, S., and Buntine, W. 1992. Machine invention of first-order predicates by inverting resolution. *Inductive logic programming* 261–280.

Platanios, A.; Blum, A.; and Mitchell, T. M. 2014. Estimating Accuracy from Unlabeled Data. In *In Proceedings of UAI 2014*.

Pujara, J.; Miao, H.; Getoor, L.; and Cohen, W. 2013. Knowledge graph identification. In *ISWC 2013*.

Samadi, M.; Veloso, M. M.; and Blum, M. 2013. Openeval: Web information query evaluation. In *AAAI*.

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

Thrun, S., and Pratt, L., eds. 1998. *Learning to learn*. Norwell, MA, USA: Kluwer Academic Publishers.

Tong, S., and Koller, D. 2001. Active learning for structure in bayesian networks. In *IJCAI*.

Wang, R. C., and Cohen, W. W. 2007. Language-independent set expansion of named entities using the web. In *Proc. of ICDM*.