DOI:0 0.0000/000

Never-Ending Learning

By T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling

Abstract

Whereas people learn many different types of knowledge from diverse experiences over many years, and become better learners over time, most current machine learning systems are much more narrow, learning just a single function or data model based on statistical analysis of a single data set. We suggest that people learn better than computers precisely because of this difference, and we suggest a key direction for machine learning research is to develop software architectures that enable intelligent agents to also learn many types of knowledge, continuously over many years, and to become better learners over time. In this paper we define more precisely this *never-ending learning* paradigm for machine learning, and we present one case study: the Never-Ending Language Learner (NELL), which achieves a number of the desired properties of a never-ending learner. NELL has been learning to read the Web 24hrs/day since January 2010, and so far has acquired a knowledge base with 120mn diverse, confidence-weighted beliefs (e.g., *servedWith(tea,biscuits)*), while learning thousands of interrelated functions that continually improve its reading competence over time. NELL has also learned to reason over its knowledge base to infer new beliefs it has not yet read from those it has, and NELL is inventing new relational predicates to extend the ontology it uses to represent beliefs. We describe the design of NELL, experimental results illustrating its behavior, and discuss both its successes and shortcomings as a case study in never-ending learning. NELL can be tracked online at http://rtw.ml.cmu.edu, and followed on Twitter at @CMUNELL.

1. INTRODUCTION

Machine learning is a highly successful branch of Artificial Intelligence (AI), and is now widely used for tasks from spam filtering, to speech recognition, to credit card fraud detection, to face recognition. Despite these successes, 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 machine learning applications, 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, reinforcement learning) 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. For example, humans first learn to crawl, then to walk, run, and perhaps ride a bike. They also learn to recognize objects, to predict their motions in different circumstances, and to control those motions. Importantly, they learn *cumulatively*: as they learn one thing this new knowledge helps them to more effectively learn the next, and if they revise their beliefs about the first then this change refines the second.

The thesis of our research 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 NELL which implements several of these capabilities, and which has been learning to read the Web 24hrs/day since January 2010, and (3) identify from NELL's strengths and weaknesses a number of key design features important to any never-ending learning system. This paper is an elaboration and extension to an earlier overview of the NELL system.[27]

2. RELATED WORK

Previous research has considered the problem of designing machine learning agents that persist over long periods

The original version of this paper appeared in the *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (Austin, TX, Jan. 25–30, 2015), 2302–2310.

of time (e.g., life long learning[38]), and that learn to learn[39] by various methods, including using previously learned knowledge from earlier tasks to improve learning of subsequent tasks.[11] Still, there remain few if any working systems that demonstrate this style of learning in practice. General architectures for problem solving and learning (e.g., SOAR Laird et al.[21], ICARUS Langley et al.[22], PRODIGY Donmez and Carbonell[15], and THEO Mitchell et al.[26]) 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 Automated Mathematician (AM) and Eurisko[24] 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[10] suggests mechanisms by which learning of one type of knowledge can guide learning of another type. Work on active and proactive learning[15, 40] and on exploitation/exploration tradeoffs[5] presents strategies by which learning agents can collect optimal training data from their environment. Work on learning of latent representations[2, 29] 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[3] explores potential synergies across sets or sequences of learning tasks. Theoretical characterizations of cotraining[4] and other multitask learning methods[1, 32] have provided insights into when and how the sample complexity of learning problems can be improved via multitask learning.

There is also related work on constructing large knowledge bases on the Web—the application that drives our NELL case study. The WebKB project,[13] Etzioni's early[17] and more recent[18] work on machine reading the Web, and the YAGO[37] project all represent attempts to construct a knowledge base using Web resources, as do commercial knowledge graph projects at Google, Yahoo!, Microsoft, Bloomberg and other companies. However, unlike NELL, none of these efforts has attempted a sustained never ending learning approach to this problem.

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.[8], 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.

3. 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 *performance 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.

$$\mathscr{L} = (L, C) \qquad (1)$$
$$where, L = \{\langle T_i, P_i, E_i \rangle\}$$
$$C = \{\langle \phi_k, V_k \rangle\}$$

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 inter-related knowledge*; that is, L contains many learning tasks, coupled by many cross-task constraints,
- *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 $\mathscr{L}$.

4. CASE STUDY: NEVER ENDING LANGUAGE LEARNER

The Never-Ending Language Learner (NELL), an early prototype of which was reported in Carlson et al.[8], is a learning agent whose task is to learn to read the Web. The input-output specification of NELL's never ending learning problem is:

Given:

- an initial ontology defining hundreds of categories (e.g., Sport, Athlete) and binary relations that hold between members of these categories (e.g., AthletePlaysSport(x,y)),
- approximately a dozen labeled training examples for each category and each relation (e.g., examples of Sport might include the noun phrases "baseball" and "soccer"),
- the Web : an initial 500mn Web pages from the ClueWeb 2009 collection,[7] augmented in 2017 by the addition of the ClueWeb 2012 collection[6] to form a collection of 1.233bn Web pages. In addition, Google has granted NELL access to 100,000 Google Application Program Interface (API) search queries each day.
- occasional interaction with humans (e.g., through NELL's public Website http://rtw.ml.cmu.edu);

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

- 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,
- learn to read better than the previous day.

NELL has been running non-stop 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 approximately 120mn interconnected beliefs (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.

5. NELL'S NEVER ENDING LEARNING PROBLEM

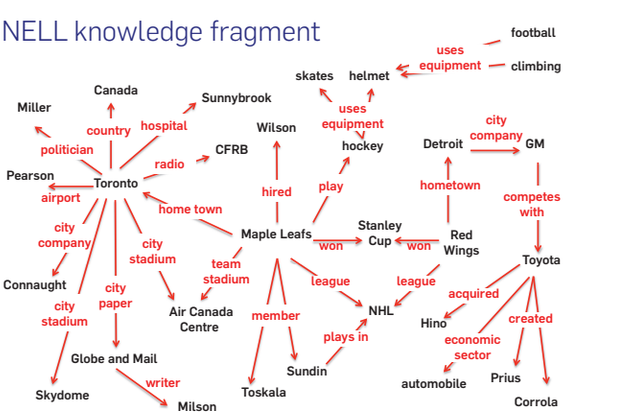
Above we described the input-output specification of the NELL system. Here we describe NELL's never-ending learning problem $\langle L, C \rangle$ in terms of the general formalism introduced in Section 2, first describing NELL's learning tasks L, then its coupling constraints C. The subsequent section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

5.1. 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 4100 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 293 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 at least five, and in some cases six distinct functions that predict Y_i, based on five different views of the noun phrase (different X_i's), which are:

Figure 1: Fragment of the 120mn 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.

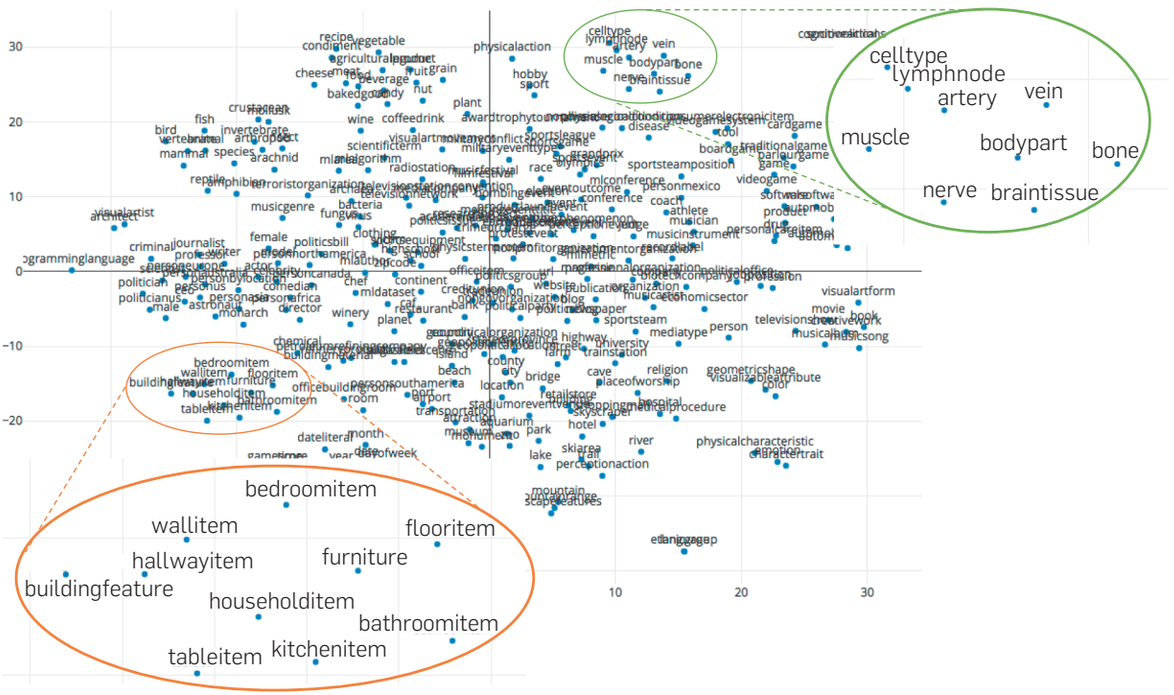


- *Character string features* of the noun phrase (e.g., whether the noun phrase ends with the character string "...burgh"). This is performed by the Coupled Morphological Classifier (CMC) system,[8] which represents the noun phrase by a vector with thousands of string features.
- *The distribution of text contexts found around this noun phrase in 1.233bn English Web pages* from the ClueWeb2009 and ClueWeb2012 text corpus (e.g., how frequently the noun phrase *N* occurs in the context "mayor of *N*"). This is performed by the Coupled Pattern Learner (CPL) system.[9]
- *The distribution of text contexts found around this noun phrase through active Web search*. This is performed by the OpenEval system,[36] which uses somewhat different context features from the above CPL system, and uses real time Web search to collect this information.
- *Hypertext Markup Language (HTML) structure of Web pages that mention the noun phrase* (e.g., whether the noun phrase is mentioned inside an HTML list, alongside other known cities). This is performed by the Set Expander for Any Language (SEAL) system.[41]
- *Visual images* associated with this noun phrase, when the noun phrase is given to an image search engine. This is performed by the Never Ending Image Learner (NEIL) system,[12] and applies only to a subset of NELL's ontology categories (e.g., not to non-visual categories such as MusicGenre).
- *Learned vector embeddings* of the noun phrases. This is performed by the Learned Embeddings (LE) module which has not been previously described, so we summarize the approach in some detail here. LE[44] learns a vector embedding for each noun phrase associated with each NELL entity, a vector embedding for each of the 293 categories in NELL's ontology, and a matrix embedding to represent the Generalizations relation that relates each NELL entity to the general categories to which it belongs. We employ a neural network architecture to learn these vector and matrix embeddings, training them on each NELL iteration to maximize their fit to the beliefs in NELL's current knowledge base. Note this knowledge base is updated on each NELL iteration in response to the combined results of all of NELL's reading and inference modules. Specifically, LE quantifies its confidence in the assertion that Generalization(X, Y) using the scoring function: $S(X_i, Y_i) = \mathbf{v}_{X_i}^T \mathbf{M} \mathbf{v}_{Y_i}$, where $\mathbf{v}_{X_i}$ and $\mathbf{v}_{Y_i}$ are d-dimensional vectors representing noun phrase X_i and NELL category Y_i respectively, and where M is a $d \times d$ matrix representing the Generalization relation. The vector embedding $\mathbf{v}_{X_i}$ is constructed by first averaging the vectors of the words in the noun phrase, then concatenating to this vector the word vector of its head noun. These vectors are initialized with pre-trained vectors for each word, obtained from Wieting et al.[42] These word vectors are then fine tuned during training, and used to produce the noun phrase vectors as described above. During training, LE minimizes a ranking loss $\max\{0, 1 - S(X_i, Y_i) + S(X_i, Y_i')\}$ for each positive training example $\langle X_i, Y_i \rangle$, paired with a negative training example $\langle X_i, Y_i' \rangle$. Positive examples are high confidence beliefs in NELL's knowledge base, and negative examples are constructed by changing the value of Y to form a belief triple which is not in NELL's knowledge base.

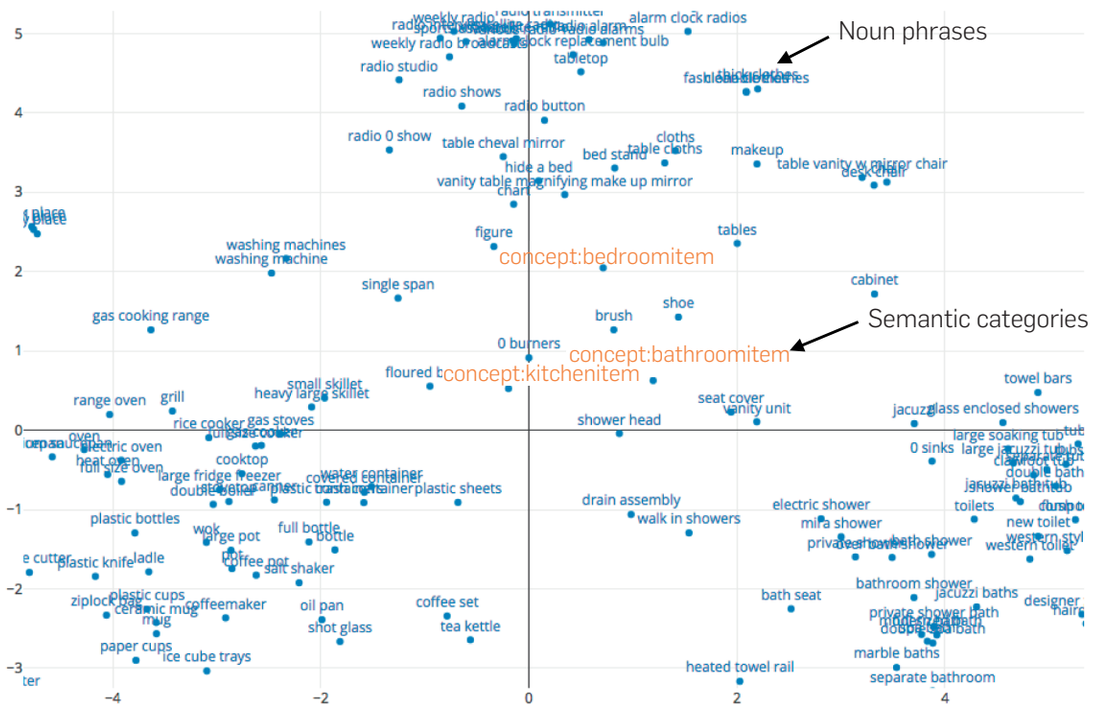
  Learned Embeddings obtains an top-1 accuracy of 0.88 when classifying new noun phrases into NELL's 293 categories. Figure 2 shows a visualization of the learned embeddings using t-SNE.[25] In general, the learned embeddings nicely reflect the semantics of the noun phrases and categories. Figure 2a displays the embeddings of 280 categories in NELL. We can see that semantically similar categories tend to be close to each other. For example, there is a cluster about body parts (on the top) and a cluster about room items (in the bottom). Figure 2b further shows three specific room-item categories and the noun phrases surrounding them. We can see that items belonging to kitchens, bedrooms, and bathrooms are generally well separated. We also find that items that can belong to multiple categories tend to locate on the boundaries. For example, "brush" and "shoe" could be both a bedroom item and a bathroom item.

- *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 461 relations in its ontology. For each relation, NELL learns four 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. These methods are described above in the list of Category Classification methods. NELL furthermore uses the LE module described above to learn to predict relation instances from learned vector embeddings of noun phrases and learned matrix embeddings for relations. Above we described the algorithm used by LE to learn vector embeddings for NELL entities and for NELL categories, and to learn a matrix embedding for the $\langle x_1$, Generalization, $x_2 \rangle$ relation. The same algorithm is used to learn a matrix embedding M_r for each NELL relation r. As in the case of the Generalization relation, LE then assigns a confidence score $S(\langle x_1, r, x_2 \rangle)$ to each possible relation triple according to the formula $S(\langle x_1, r, x_2 \rangle) = \mathbf{v}_{x_1}^T \mathbf{M}_r \mathbf{v}_{x_2}$, where $\mathbf{v}_{x_1}$ and $\mathbf{v}_{x_2}$ represent the learned vector embeddings for x_1 and x_2, and where $\mathbf{M}_\mathbf{r}$ is the learned matrix embedding for relation r. In general we find LE's inferences about relations other than Generalization are less accurate than for the Generalization relation.

Figure 2: t-SNE visualization of the embeddings learned by LE.



(a) Embeddings of the semantic categories.



(b) Embeddings of the noun phrases and semantic categories.

This may be due in part to the smaller number of training examples for these relations, and may in part be due to the greater suitability for our approach to semantic category assignment (i.e., predicting the Generalization) compared to predicting other relations such as PersonFoundedCompany().

- *Entity Resolution*: Functions that classify whether pairs of noun phrases are synonyms. NELL's knowledge base represents noun phrases as distinct from the entities to which they can refer. This is essential because polysemous words can refer to multiple types of entities (e.g., the word "coach" can refer to a type of "person," or a type of "vehicle"), and because synonymous words can refer to the same entity (e.g., "NYC," "New York City" and "Big Apple" are synonyms for the same entity). In order to deal with polysemy, NELL simply allows a noun phrase to be classified into multiple categories if there is strong evidence according to its reading methods. To deal with synonymy, NELL learns explicit 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[20]. For each of NELL's 293 categories, it co-trains two synonym classifiers. One classifier is based on string similarity between the two noun phrases (e.g., "NYC" and "New York City" have similar string features). The second is based on similarities in the beliefs NELL has extracted (e.g., if NELL's KB believes that "NYC" and "New York City" have the same mayor, this is evidence that the assumed two city names may be synonyms, though belonging to the same country might not be evidence that two city names are synonyms). NELL learns for each of its categories (e.g., "city"), what are the category specific types of knowledge that are evidence of synonymy, and which types of string features indicate synonymy.
- *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 Path Ranking Algorithm (PRA) system.[19, 23]

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.

5.2. 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. We describe them below as hard logical constraints. However, NELL uses these primarily as soft constraints that can be violated at some penalty cost.

- *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,[4] 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 types of its arguments must be defined in terms of NELL categories (e.g., "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 4100 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 inventing new predicates for its ontology (as discussed below).

6. NELL'S LEARNING METHODS AND ARCHITECTURE

The software architecture for NELL, depicted in Figure 3, includes a KB which acts as a blackboard through which NELL's various learning and inference modules communicate.[a] As shown in Figure 3, these software modules map closely to the learning methods (CPL, CMC, SEAL, OpenEval, PRA, and NEIL) for the different types of functions mentioned in the previous section, so that NELL's various learning tasks are partitioned across these modules.

6.1. Learning in NELL as an Approximation To EM

NELL is in an infinite loop analogous to an Expectation-Maximization (EM) algorithm[14, 30] for semi-supervised learning, performing an E-like step and an M-like step on each iteration through the loop. During the E-like step, the set of beliefs that form the knowledge base is re-estimated; that is, each reading and inference module in NELL 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 reading and inference modules, employing module-specific learning algorithms for each. 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, and to delete beliefs in which it subsequently loses confidence. In addition, NELL enforces its consistency constraints in a limited-radius fashion on each iteration (i.e., if one belief changes, the only other beliefs influenced are those coupled directly by some constraint; compositions of constraints are not considered). However, across multiple iterations of its EM-like algorithm, the influence of any given belief update can propagate throughout the knowledge graph as neighboring beliefs are themselves revised.

6.2. Knowledge Integrator in NELL

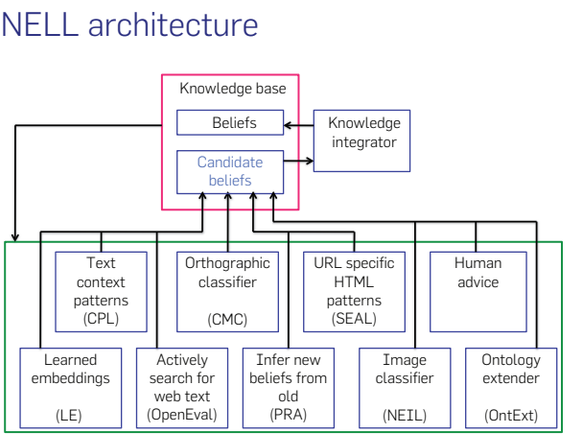
The KI integrates the incoming proposals for KB updates. For efficiency, the KI considers only moderate-confidence candidate beliefs, and re-assesses confidence using a limited radius subgraph of the full graph of consistency constraints and beliefs. As an example, for each new relational triple that the KI asserts, it checks that the entities in the relational triple have a category type consistent with the relation, but does not consider using new triples as a trigger to update beliefs about these argument types during the same iteration. Over multiple iterations, the effects of constraints propagate more widely through this graph of beliefs and constraints. In Pujara et al.[35] a more effective algorithm is proposed for the joint inference problem faced by NELL's KI; we believe it will be helpful to upgrade NELL's KI in the future to use this approach.

6.3. 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.[28]

[a] 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's software architecture. NELL's growing knowledge base (red box) serves as a shared blackboard through which its various reading and inference modules (green box) interact. On each NELL iteration the knowledge base is first updated by integrating proposals from the various reading and inference modules. The revised knowledge base is then used to retrain each of these modules.



OntExt searches for new relations by considering every pair of categories in NELL's current ontology, to search for evidence of a new, frequently discussed relation between members of that category pair. It performs this search in a three step process: (1) Extract sentences mentioning known instances of both categories (e.g., for the category pair ⟨*drug,disease*⟩ 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; a sample is shown in Figure 4. Note the invention and introduction of each new relation into NELL's ontology spawns a number of new tasks. These include new "learning to read" tasks, to classify which noun phrase pairs satisfy the relation, based on different views of the noun phrase pair. Each new relation also spawns a new task of learning Horn clause rules to infer this new relation from others, and of course the new relation also becomes available for representing new rules that infer instances of other NELL relations.

In addition to the OntExt algorithm for proposing new relations, we have more recently developed the verb knowledge base (VerbKB[b]) which proposes new relations on a much larger scale.[43] Verbs and verb phrases naturally express relations between noun phrases, and can provide the high coverage vocabulary of relation predicates required to represent beliefs in arbitrary text. VerbKB groups semantically similar verb patterns by analyzing the statistics of all subject, verb lexeme (plus preposition where available), and object triples (e.g., *<horse, eat, hay>, <john, eat with, fork>*) found by parsing the 500 mn English Web pages in NELL's initial cache of Web pages from ClueWeb2009.

VerbKB is guided by NELL's knowledge about the semantic categories to which the subject and object belong. The groups of *<subjectCategory verbCluster, objectCategory>* patterns discovered by VerbKB are proposed as new typed relations for NELL. For example, VerbKB has proposed that the group of verbs {have, experience, suffer, survive, sustain, bear, endure, tolerate} represents a potential new NELL relation PersonHaveDisease(person,disease), when these verbs occur with a subject belonging to the NELL category "Person" and an object of NELL category "Disease." VerbKB has clustered 65,000 verb lexemes (+prepositions) which cover 98% of all verb mentions in ClueWeb2010 and has proposed a collection of 86,000 verb clusters (58,000 of which are non-singleton clusters) as new relations to NELL. Because this very large number of proposed relations raises scaling issues for NELL's current hardware and software, we are currently exploring ways to scale up NELL, and ways to select among these proposed relations by relying on NELL's Twitter interface[c] followers to decide (as described in Pedro and Hruschka[31]) which among these relations will be most interesting for NELL to learn. Although we are still working to incorporate this into routine use by NELL's never-ending execution run, we are optimistic that this will provide a significant increase in the coverage and capability of NELL's learned knowledge.

6.4. Self Reflection and Self Evaluation

One important capability we wish to add to NELL is the ability to self-reflect on, and self-evaluate its own performance, to enable it to focus its learning efforts where it most needs improvement. Although NELL's architecture does not yet have such a self-reflection component, we have recently developed and tested the key algorithms that will enable it to estimate the accuracies of thousands of functions it is learning, based solely on the unlabeled data it has access to. The key theoretical question here is "under what conditions can unlabeled data be used to estimate accuracy of learned functions?" Surprisingly, we have found that there are conditions under which the observed *consistency* among different learned functions applied to unlabeled data can be used to derive highly precise estimates of *accuracies* of these functions, and that these methods work well for accuracy estimation in NELL.

For example, in Platanios et al.[32] we show that if one has three or more approximations to the same function (e.g., NELL's different learned classifiers that predict whether a noun phase refers to a city, based on different views of the noun phrase), if these functions are more accurate than chance, and if their errors are independent, then the rates at which these functions agree on the classification of unlabeled examples can be used to solve exactly for their accuracies. While NELL comes close to meeting

[b] http://gourierverb.azurewebsites.net.

[c] https://twitter.com/cmunell.

Figure 4: Sample of relations automatically discovered by NELL's OntExt algorithm. When NELL adds a newly discovered relation to its ontology, its learning algorithms are automatically triggered to seek new instances. For example, since adding these new relations NELL has added hundreds of instances of buildingFeatureMadeFromMaterial including (tiles, porcelain) and (garage doors, steel), and thousands of instances of clothingGoesWithClothing including (tee shirt, jeans), (tuxedo jacket, tie) and (gloves, warm coat).

Sample of self-discovered NELL relations

- athleteWonAward
- animalEatsFood
- languageTaughtInCity
- clothingMadeFromPlant
- beverageServedWithFood
- fishServedWithFood
- athleteBeatAthlete
- athleteInjuredBodyPart
- arthropodFeedsOnInsect
- animalEatsVegetable
- plantRepresentsEmotion
- foodDecreasesRiskOfDisease
- clothingGoesWithClothing
- bacteriaCausesPhysCondition
- buildingFeatureMadeFromMaterial
- emotionAssociatedWithDisease
- foodCanCauseDisease
- agriculturalProductAttractsInsect
- arteryArisesFromArtery
- countryHasSportsFans
- bakedGoodServedWithBeverage
- beverageContainsProtein
- animalCanDevelopDisease
- beverageMadeFromBeverage

these conditions, in general it does not satisfy the assumption that its different functions make completely independent errors. However, we found it possible to weaken the assumption of independent errors, and effectively replace it by a prior stating that more independent errors are more probable. Experimental results show that these algorithms, run on NELL's learned functions for 15 representative categories, yield accuracy estimates that deviate on average less than 0.01 from the true accuracies. In Platanios et al.[33] we introduce a related Bayesian approach which also leverages the fact that NELL is learning to predict many different functions for each input noun phrase, hence leveraging the full mulit-view, multi-task nature of NELL's learning problem. Finally, in Platanios et al.[34] we also propose a probabilistic logic approach that further leverages the information provided by logical constraints between the outputs of the functions that NELL is learning to predict (e.g., a noun phrase that refers to a city has to also refer to a location).

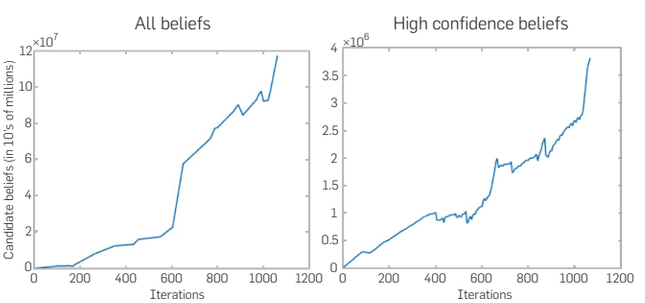
7. EMPIRICAL EVALUATION

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 July 10, 2017, during which NELL completed 1064 iterations. The left panel of Figure 5 shows the number of beliefs in NELL's KB over time, and the right panel of Figure 5 shows the number of beliefs for which NELL holds high confidence. Note that as of July 2017, NELL's KB contains approximately 117mn beliefs with varying levels of confidence, including 3.81mn that 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.

As Figure 5 illustrates, NELL's KB is clearly growing, though its high confidence beliefs constitute only about 3% of the total set of beliefs it is considering. Although NELL has now saturated some of the categories and relations in its ontology (e.g., for the category "Country" it extracted most actual country names during the first few hundred iterations), the knowledge base nevertheless continues to grow overall. This ongoing growth is due in part to the fact that NELL's ontology extension module is adding new predicates to the ontology over time (e.g., *athleteInjuredBodyPart(athlete, bodyPart)*), creating the opportunity for NELL to acquire new beliefs that it could not even represent in its original ontology.

Figure 5: 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. The horizontal axis covers NELL iterations from January 2010 until July 2017.



Beyond the volume of beliefs, 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 its cache of English Web pages, plus the world wide Web as accessed through NELL's reading modules. 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 from the Web, then had it apply its trained methods to a test 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.[d]

In greater detail, we first selected 12 different points in time to test NELL's reading competence: iterations 166, 261, 337, 447, 490, 561, 641, 731, 791, 886, 960, and 1026. These iterations span from the inception of NELL in January 2010 through November 2016. 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 initial ontology. Each iteration-specific trained version of NELL was then applied to produce a ranked list of the top 1000 *novel* predictions, omitting any prediction corresponding to a noun phrase or relation instance for which NELL had received human feedback at any point in its history. To estimate NELL's reading competence at each point we first created a pool of test instances to manually annotate. For each iteration to be evaluated, this pool included the top 10 ranked predictions for each predicate, 20 more predictions sampled uniformly at random from ranks 11 to 100, and an additional 20 from ranks 101 to 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's learned reading competence at each iteration.

The results of this evaluation are summarized in Figure 6, which shows the improvement in NELL's reading competence over time, as measured by NELL's estimated Mean

[d] To be precise, for NELL iterations prior to and including November 2014, we used test data from the Web as of November 2014. For evaluations of NELL's competence in November 2015 and November 2016, we instead used the Web as it existed on those dates.

Average Precision (MAP) over this sample of 1000 most confident predictions for each of these 31 predicates. Taken together, the results in the Figure 6 and the results of Figure 5 show that over several years NELL's reading accuracy, and the accuracy of its most confident beliefs have grown at the same time that the volume of beliefs in the knowledge base has also grown by millions.

Next, we summarize feedback from humans to NELL. This feedback is nearly all negative feedback identifying NELL's incorrect beliefs. Figure 7 shows the distribution of this negative feedback from humans to NELL over its first 802 iterations, which is very similar to the distribution of feedback in more recent iterations. During this period, NELL received on average 2.4 negative feedback labels per predicate, per month, for a total of 85,088 items of negative feedback (an average of 1,467 per month). 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 most of the feedback is now crowdsourced, that is, provided by external visitors to the NELL Website, or by followers of @CMUNELL on Twitter.

Figure 6: Evolution of NELL reading accuracy over time. The vertical axis shows the estimated Mean Average Precision over the 1000 most confident predictions for a representative sample of 18 categories and 13 relations in NELL's ontology. The horizontal axis represents NELL iterations from January 2010 through November 2016.

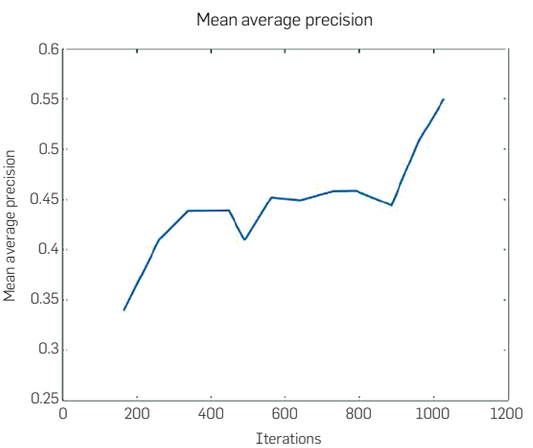
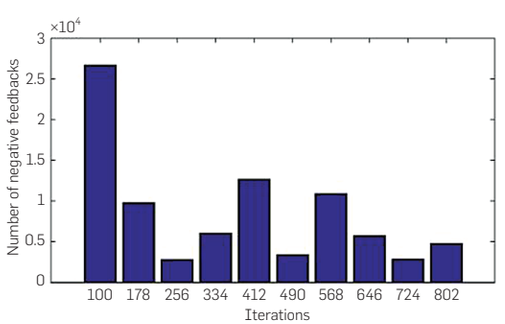


Figure 7: Human feedback to NELL over time. Each bar in this histogram shows the number of NELL beliefs for which humans provided negative feedback, during a 78 iteration interval. This averages out to 2.4 items of feedback per month, per predicate in NELL's ontology. Human input to NELL has been dropping over time, even as its reading accuracy has been increasing.



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 whether it has other mutually exclusive categories to learn—this mutually exclusive relationship provides a coupling constraint that typically yields valuable 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 to its ontology other 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 performs poorly for a predicate due to a particular error which propagates due to its bootstrap-style learning from unlabeled or self-labeled data. 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.

It is important to realize that as NELL progresses, the task of adding the next new belief to the knowledge base naturally becomes more difficult. NELL's redundancy-based reading methods tend to extract the most frequently-mentioned beliefs earlier (e.g., for the category "emotions" NELL first extracted frequently mentioned emotions such as "gladness" and "loneliness"). But once it has extracted the frequently mentioned instances which are easiest for its statistically-based methods, it later can only grow the KB by extracting less frequently mentioned beliefs (e.g., later the emotions it was able to add were more obscure instances such as "incredible lightness," "cavilingness," and "nonop-probriousness," as well as some non-emotion phrases).

This increasing difficulty over time seems to be inherent to the task of never-ending learning. Meeting this challenge in NELL 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,[16] image data as well as text, and new continuous streams of data such as 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.

8. DISCUSSION

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. In this paper, we present NELL as an early case study of a never-ending learning system. What are the lessons to be learned from this case study? Our experience with NELL suggests four useful design features that have led to the successes it has had—design features 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 hundreds of thousands of probabilistic Horn clauses and related probabilistic inference rules 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 joint 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—that is, 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; for example, it 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. In addition, recent research in natural language has shown that working wth non-symbolic vector embeddings of words, phrases and entities, learned via deep neural networks, has many advantages. In NELL, the recent addition of the LE method has similarly yielded improvements in NELL's ability to extract new instances of categories and relations. However, an even more dramatic adoption of vector embeddings learned via deep networks would be possible, for example, providing a continuous space of category and relation predicates represented by vectors and matrices, fundamentally changing the framing of the ontology extension problem (i.e., if every relation is represented by a matrix, the set of possible matrices *is* the set of possible relations in the ontology).

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 and correct perceptions, or incorrect predictions and correspondingly 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 et al.[32] 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, etcetera. 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, and thank Lucas Navarro, Bill McDowell, Oscar Romero and Amos Azaria for help in the empirical evaluation of NELL. This research was supported in part by DARPA under contract number FA8750-13-2-0005, by NSF grants IIS-1065251 and CCF-1116892, by AFOSR award FA9550-17-1-0218, and by several generous gifts from Google. We also gratefully acknowledge graduate fellowship support over the years from Yahoo, Microsoft, Google, Fulbright, CAPES, and FAPESP.

References

1. Balcan, M.-F., Blum, A. A PAC-style model for learning from labeled and unlabeled data. *Proc. of COLT* (2004).
2. Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning 2*, 1 (2009), 1–127.
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (2009), ACM, 41–48.
4. Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training. *Proc. of COLT* (1998).
5. Brunskill, E., Leffler, B., Li, L., Littman, M.L., Roy, N. Corl: A continuous-state offset-dynamics reinforcement learner. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)* (2012), 53–61.
6. Callan, J. Clueweb12 data set (2013) http://lemurproject.org/clueweb12/.
7. Callan, J., Hoy, M. Clueweb09 data set (2009) http://boston.lti.cs.cmu.edu/Data/clueweb09/.
8. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E.R., Mitchell, T.M. Toward an architecture for never-ending language learning. *AAAI 5*, 3 (2010a).
9. Carlson, A., Betteridge, J., Wang, R.C., Hruschka Jr., E.R., Mitchell, T.M. Coupled semi-supervised learning for information extraction. *Proc. of WSDM* (2010b).
10. Caruana, R. Multitask learning. *Machine Learning 28* (1997), 41–75.
11. Chen, Z., Liu, B. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 10*, 3 (2016), 1–145.
12. Chen, X., Shrivastava, A., Gupta, A. Neil: Extracting visual knowledge from web data. *In Proceedings of ICCV* (2013).
13. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the 15th National Conference on Artificial Intelligence* (1998).
14. Dempster, A., Laird, N., Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Series B* (1977).
15. Donmez, P., Carbonell, J.G. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management* (2008), ACM, 619–628.
16. Duarte, M.C., Hruschka Jr., E.R. How to read the web in portuguese using the never-ending language learner's principles. In *Intelligent Systems Design and Applications (ISDA), 2014 14th International Conference on* (2014), IEEE, 162–167.
17. Etzioni, O.e.a. Web-scale information extraction in knowitall (preliminary results). In *WWW* (2004).
18. Etzioni, O.e.a. Open information extraction: The second generation. *Proc. of IJCAI* (2011).
19. Gardner, M., Talukdar, P., Krishnamurthy, J., Mitchell, T. Incorporating vector space similarity in random walk inference over knowledge bases. *Proc. of EMNLP* (2014).
20. Krishnamurthy, J., Mitchell, T.M. Which noun phrases denote which concepts. *Proc. of ACL* (2011).
21. Laird, J., Newell, A., Rosenbloom, P. SOAR: An architecture for general intelligence. *Artif. Intel. 33*, (1987), 1–64.
22. Langley, P., McKusick, K.B., Allen, J.A., Iba, W.F., Thompson, K. A design for the ICARUS architecture. *SIGART Bull. 2*, 4 (1991), 104–109.
23. Lao, N., Mitchell, T., Cohen, W.W. Random walk inference and learning in a large scale knowledge base. *Proc. of EMNLP* (2011).
24. Lenat, D.B. Eurisko: A program that learns new heuristics and domain concepts. *Artif. Intel. 21*, 1–2 (1983), 61–98.
25. Maaten, L.v.d., Hinton, G. Visualizing data using t-SNE. *J. Machine Learning Res. 9*, Nov (2008):2579–2605.
26. Mitchell, T.M., Allen, J., Chalasani, P., Cheng, J., Etzioni, O., Ringuette, M.N., Schlimmer, J.C. THEO: A framework for self-improving systems. *Arch. for Intel.* (1991), 323–356.
27. Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N.,

Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., Welling, J. Never-ending learning. In *AAAI Conference on Artificial Intelligence* (2015), AAAI, 2302–2310.
28. Mohamed, T., Hruschka Jr., E.R., Mitchell, T.M. Discovering relations between noun categories. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (2011), Association for Computational Linguistics, , Edinburgh, Scotland, UK, 1447–1455.
29. Muggleton, S., Buntine, W. Machine invention of first-order predicates by inverting resolution. *Inductive logic programming* (1992), 261–280.
30. Nigam, K., McCallum, A., Thrun, S., Mitchell, T. Text classification using labeled and unlabeled documents. *Machine Learning 39* (2000), 103–134.
31. Pedro, S.D., Hruschka Jr, E.R. Conversing learning: Active learning and active social interaction for human supervision in never-ending learning systems. In *Advances in Artificial Intelligence–IBERAMIA 2012* (Springer, 2012), 231–240.
32. Platanios, E.A., Blum, A., Mitchell, T.M. Estimating Accuracy from Unlabeled Data. *Proc. of UAI* (2014).
33. Platanios, E.A., Dubey, A., Mitchell, T.M. Estimating Accuracy from Unlabeled Data: A Bayesian Approach. In *Proceedings of the International Conference on Machine Learning* (2016).
34. Platanios, E.A., Poon, H., Mitchell, T.M., Horvitz, E. Estimating Accuracy from Unlabeled Data: A Probabilistic Logic Approach (2017). preprint, https://arxiv.org/abs/1705.07086.
35. Pujara, J., Miao, H., Getoor, L., Cohen, W. Knowledge graph identification. *ISWC* (2013).
36. Samadi, M., Veloso, M.M., Blum, M. Openeval: Web information query evaluation. In *AAAI* (2013).
37. Suchanek, F.M., Kasneci, G., Weikum, G. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)* (2007), ACM Press, New York, NY, USA.
38. Thrun, S., Mitchell, T. Lifelong robot learning. *Rob. Auton. Sys. 15*, (1995), 25–46.
39. Thrun, S., Pratt, L. (eds) *Learning to learn*, Kluwer Academic Publishers, Norwell, MA, USA, 1998.
40. Tong, S., Koller, D. Active learning for structure in bayesian networks. *IJCAI* (2001).
41. Wang, R.C., Cohen, W.W. Language-independent set expansion of named entities using the web. *Proc. of ICDM* (2007).
42. Wieting, J., Bansal, M., Gimpel, K., Livescu, K. Towards universal paraphrastic sentence embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR)* (2015).
43. Wijaya, D.T. *VerbKB: A Knowledge Base of Verbs for Natural Language Understanding*. Ph.D. Dissertation, Carnegie Mellon University, 2016.
44. Yang, B., Mitchell, T. Leveraging knowledge bases in lstms for improving machine reading. *ACL* (2017).

T. Mitchell (tom.mitchell@cs.cmu.edu), Carnegie Mellon University, USA.

W. Cohen, B. Yang, J. Betteridge, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, K. Mazaitis, N. Nakashole, E. Platanios, M. Samadi, D. Wijaya, A. Gupta, X. Chen, and A. Saparov, Carnegie Mellon University, USA.

E. Hruschka, Federal University of São Carlos, Brazil.

P. Talukdar, Indian Institute of Science, India.

A. Carlson and N. Lao, Google Inc., USA.

T. Mohamed and R. Wang, Research carried out while at Carnegie Mellon University.

A. Ritter, Ohio State University, USA.

B. Settles, Duolingo, USA.

M. Greaves, Alpine Data Labs, USA.

J. Welling, Pittsburgh Supercomputing Center, USA.