Toward Never Ending Language Learning

Justin Betteridge[1], **Andrew Carlson**[1], **Sue Ann Hong**[1], **Estevam R. Hruschka Jr.**[1,2],
Edith L. M. Law[1], **Tom M. Mitchell**[1] and **Sophie H. Wang**[1]
[1]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
[2]Federal University of Sao Carlos
Sao Carlos, SP - Brazil

Abstract

We report research toward a never-ending language learning system, focusing on a first implementation which learns to classify occurrences of noun phrases according to lexical categories such as "city" and "university." Our experiments suggest that the accuracy of classifiers produced by semi-supervised learning can be improved by coupling the learning of multiple classes based on background knowledge about relationships between the classes (e.g., "university" is mutually exclusive of "company", and is a subset of "organization").

Introduction

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

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

The thesis underlying this research goal is that the vast redundancy of information on the web (e.g., many facts are stated multiple times with different words) will enable a system with the right learning mechanisms and capabilities for self-reflection to learn with only minimal supervision. We suggest that research toward never-ending language learning holds the potential to yield major steps forward in the state of the art of natural language understanding.

This abstract describes preliminary experiments with a prototype system called NELL (Never-Ending Language Learner). At present, NELL acquires two types of natural language processing capability: identifying noun phrases that are members of specific semantic *classes*, such as cities, companies, and universities, and identifying pairs of noun phrases that are members of specific semantic *relations*, such as "`<org>` is headquartered in `<loc>`". The focus of this abstract, however, is on semantic classes.

With this system, we focus on an important aspect of our broad research goals: the problem of self-supervised learning from primarily unlabeled data. In particular, we demonstrate a learning method that uses knowledge about subset and mutual exclusion relations between classes to couple the semi-supervised learning of these classes in order to improve accuracy of learning.

Related Work

Our work is closely related to the KnowItAll system (Etzioni and others 2005). However, our project has a different focus. Our aim is to build a system that can learn continuously, and discovering members of semantic categories is an application. In particular, NELL is designed to learn many classifiers at once from primarily unlabeled data, coupling the learning of these classifiers in order to improve accuracy.

We build on the work of Riloff and Jones (1999), which iteratively learned patterns and instances of semantic categories. Here again, the coupling of multiple learning tasks is our main extension to this previous work.

System Description

The input to NELL is an initial *ontology* $\mathcal{O}$, with four types of pre-specified information for each class: (1) some noun phrases that are trusted *instances* of that class, (2) a few trusted contextual *patterns* that are known to yield high-precision in extraction (e.g. "cities such as _")[1], (3) a list of other classes in the ontology with which that class is mutually exclusive and (4) a list of other classes of which that class is a subset. NELL's main goal is to grow its list of trusted instances for each class as much as possible with high precision. In pursuance of this goal, NELL also incrementally expands each class's set of trusted patterns.

Algorithm 1 gives a summary of NELL's approach. To start, the seed instances/patterns are shared among classes according to pre-defined relationships. For example, if class A is *mutually exclusive* with class B, A's trusted instances/patterns become *negative* instances/patterns for B, and vice versa. However, if A is a *subset* of B, A's trusted instances/patterns become trusted items for B. Then, for an indefinite number of iterations, we expand the sets of trusted instances/patterns for each class.

First, NELL generates new candidate instances by using each trusted pattern as a query and extracting the noun phrases that co-occur with the query pattern in a web corpus of 8 million web pages. For scalability reasons, we limit extraction to 500 candidates. NELL then filters out candidate

[1] we refer to these pre-specified instances/patterns as *seeds*

Algorithm 1: The main loop of NELL.

```
Input: An ontology $\mathcal{O}$
Output: Trusted instances/patterns for each class

SHARE initial instances/patterns among classes;
for $i = 1, 2, \ldots, \infty$ do
    foreach class $c \in \mathcal{O}$ do
        EXTRACT new candidate instances/patterns;
        FILTER candidates;
        TRAIN instance/pattern classifiers;
        ASSESS candidates using trained classifiers;
        PROMOTE highest-confidence candidates;
    end
1   SHARE promoted items as PMI features;
2   SHARE promoted items as examples for
    classification;
end
```

instances that do not co-occur with at least two trusted patterns or that co-occur with any negative pattern in the same web corpus. We also extract and filter new candidate patterns using the trusted positive and negative instances in a completely analogous manner[2].

Next, for each class in the ontology NELL trains a discretized Naïve Bayes classifier[3] to classify the candidate instances. Its features include pointwise mutual information (PMI) scores (Turney 2001) of the candidate instance with each of the trusted patterns associated with the class. It also uses as a feature the log probability from a multinomial Naïve Bayes bag-of-words (BOW) classifier based on the contextual word distribution of the instance under consideration in our large web corpus. The current sets of trusted and negative instances are used to train the classifier. NELL also trains a completely analogous classifier for patterns, except with two additional features: the estimated precision and estimated recall of the pattern under consideration.

NELL uses these classifiers to assess the sets of candidate instances/patterns, then ranks the positively classified candidates according to their classification scores and promotes p of them to trusted status, where $p = \min(\text{top } r\%, 10)$[4].

At the end of each iteration, NELL couples the classification of instances/patterns for the classes in our ontology in two ways: (1) it uses all of the trusted instances/patterns from all of the classes as PMI features for training its classifiers and (2) it shares the recently promoted instances/patterns among classes using the pre-defined mutual exclusion and subset relations, just as it did initially using the seeds.

Experimental Results

We performed experiments designed to test whether our methods for exploiting the coupled nature of our learning problems do in fact improve the performance of the system.

Table shows the results of experiments comparing two versions of NELL. The version labeled **C** (for *Coupled*) refers to the system as described in Algorithm 1; **U** (for *Uncoupled*) refers to the version not performing sharing of features and promoted examples between classes, hence omitting the lines labeled 1 and 2 in Algorithm 1. We populated categories "city," "country," "company," and "university", all of which were declared mutually exclusive with each other, and we initialized the system with 15 seed instances and 3 seed patterns per class[5].

	country	city	company	university	mean
U	93.6	99.1	100.0	79.1	93.0
C	89.1	98.2	100.0	97.3	96.2

Table 1: Precision for each class on promoted instances from the uncoupled (**U**) and coupled (**C**) systems after 11 iterations. Each class had 110 promoted instances.

These initial results suggest there may be advantages to our coupling mechanisms. The greatest difference between the coupled and uncoupled versions of the system is in the "university" class, where the uncoupled version started promoting incorrect examples such as "the bottom" and "the same time" while the coupled version was still promoting reasonable examples. We see this as evidence that the coupling of classes prevents divergence in bootstrapping. The second greatest difference is observed for the "country" class, where the uncoupled version is more accurate. However, many of the errors for the coupled system come from promoting larger geographical regions, such as "Central America" and "Central Asia." If the systems were also learning a "continent" class, we would not expect to see as many of these errors.

Acknowledgments

This work is supported in part by DARPA, Google, a Lucent Fellowship to Sue Ann Hong, a Yahoo! Fellowship to Andrew Carlson, and the Brazilian research agency CNPq. We also gratefully acknowledge Jamie Callan for making available his collection of web data and Yahoo! for use of their M45 computing cluster.

References

Etzioni, O., et al. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence* 165(1):91–134.

Riloff, E., and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 474–479.

Turney, P. D. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, 491–502.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd edition.

[2] Patterns are extracted only for pre-specified syntactic forms based on the OpenNLP part-of-speech tagger.

[3] from the Weka package (Witten and Frank 2005)

[4] For these experiments, r was set to 50

[5] Seeds from 7 other mutually exclusive categories provided additional negative evidence in both configurations.