In *AAAI-09 Spring Symposium on Learning by Reading and Learning to Read*

Learning and Evaluating the Content and Structure of a Term Taxonomy

Zornitsa Kozareva
DLSI, University of Alicante
Campus de San Vicente
Alicante, Spain 03080
zkozareva@dlsi.ua.es

Eduard Hovy
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
hovy@isi.edu

Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu

Abstract

In this paper, we describe a weakly supervised bootstraping algorithm that reads Web texts and learns taxonomy terms. The bootstrapping algorithm starts with two seed words (a seed hypernym (Root concept) and a seed hyponym) that are inserted into a doubly anchored hyponym pattern. In alternating rounds, the algorithm learns new hyponym terms and new hypernym terms that are subordinate to the Root concept. We conducted an extensive evaluation with human annotators to evaluate the learned hyponym and hypernym terms for two categories: animals and people.

Motivation

It is generally accepted that Learning by Reading (LbR) systems can never start truly from scratch, knowing nothing. A certain amount of basic conceptual knowledge, in the form of a seed set of terms and perhaps an overall framework structure, has to be provided. Some frameworks that have been suggested are ontologies, term taxonomies, sets of interconnected propositions, and libraries of functions. It is then the task of the LbR system to read texts, extract and structure more information, and insert this into the framework appropriately, thereby producing additional, richer, sets of terms and/or interrelationships (which we call ‘knowledge’).

In this paper we focus on one variant of this problem: building an enhanced term taxonomy, en route toward an ontology, with both its terms and its structure justified by evidence drawn from text. We start with one hypernym term and one hyponym term (i.e., one example of a hypernym relation) and then set out to read texts, learn additional terms, and classify them appropriately. We focus solely on ISA relationships, and use a definition of ISA that allows a term (concept) to have several ISA relationships at the same time. This task is not LbR in its ‘traditional’ sense, but is a form of LbR that enables the other forms of it, since the results of this task are enhanced background knowledge.

In previous work, we developed a bootstrapping algorithm that begins with one instance of a hypernym relation (i.e., a hypernym/hyponym pair) and iteratively learns more hyponyms through a combination of web querying and graph algorithms (Kozareva, Riloff, & Hovy 2008). This process produces a list of terms that are hyponyms of the given hypernym. Having such a *semantic lexicon* is tremendously useful, but our ultimate goal is to learn a richer taxonomic structure, as automatically as possible. With this goal in mind, we have created a new bootstrapping algorithm that learns new hypernyms (superordinate terms) for a set of subordinate terms. Given a set of animal instances (e.g., *dog*, *cat*), we discover new terms that are superordinate category names (e.g., *mammal*, *pet*). By combining hypernym and hyponym learning in an alternating fashion, we can iteratively learn new hypernym/hyponym relations. In essence, our problem is to learn from reading texts all subconcepts for a Root concept, and to organize them appropriately.

Our work has forced us to confront head on the problem of evaluating the structure and contents of a term taxonomy. In this paper, we explain why this type of evaluation is so challenging and give many examples that illustrate how rich and complex category learning can be. First, we begin by presenting our previous bootstrapping algorithm to learn hyponym terms. Next, we present our new bootstrapping algorithm that alternately learns hypernym and hyponym terms. This algorithm produced a large number of hypernym category terms, and the wide-ranging nature of the terms was staggering. To better understand the nature of the category terms that were learned, we created a detailed set of annotation guidelines that classify each term based on the type of concept that it represents. We then had several human reviewers manually classify each learned term, and measured their inter-annotator agreement levels.

Background: Bootstrapped Learning of Hyponym Terms

Previously, we developed a bootstrapping algorithm (Kozareva, Riloff, & Hovy 2008) that learns hyponyms of a given concept using a *doubly-anchored hyponym pattern*. We will describe this algorithm in some detail because our new bootstrapping algorithm builds upon that work.

The *hyponym bootstrapping algorithm* begins with one term that represents the “Root” concept, and another term that is a hyponym of the Root. These terms are instantiated in a *doubly-anchored hyponym pattern* of the form:

<*hypernym*> *such as* <*hyponym*> *and* *

We call this a *hyponym pattern* because we can apply the pattern to text to acquire additional hyponyms of the hypernym, as first suggested by (Hearst 1992). The asterisk (*) indicates the location from which new terms are extracted. In contrast to to Hearst's hyponym pattern, which is instantiated only with a hypernym, our pattern is "doubly anchored" by both a hypernym and a hyponym. The doubly-anchored nature of the pattern increases the likelihood of finding a true list construction (our system does not use part-of-speech tagging or parsing) and virtually eliminates ambiguity because the *hypernym* and *hyponym* mutually disambiguate each other. For example, the word FORD could refer to an automobile or a person, but in the pattern "CARS *such as* FORD *and* *" it will almost certainly refer to an automobile. Similarly, the class "PRESIDENT" could refer to country presidents or corporate presidents, and "BUSH" could refer to a plant or a person. But in the pattern "PRESIDENTS *such as* BUSH", both words will refer to country presidents.

Our bootstrapping process begins with two seed words: a class name (the hypernym) and a class member (the hyponym). The doubly-anchored hyponym pattern is instantiated with the seed words and given to Google as a web query. Additional hyponyms are then extracted from the position of the asterisk (*). The process can be bootstrapped by replacing the seed hyponym with each of the newly learned hyponyms, in turn, and issuing additional web queries. The bootstrapping process is implemented as a breadth-first search that iterates until no new hyponyms are extracted.

Although many of the extracted words will be true hyponyms, the pattern alone is not reliable enough to produce a highly accurate set of hyponym terms. In (Kozareva, Riloff, & Hovy 2008), we present several graph algorithms that can be used to dramatically improve the accuracy of the algorithm. Here, we use the re-ranking algorithm with precompiled *Hyponym Pattern Linkage Graphs*. When the search terminates, we have a large set of candidate hyponym terms so a graph is constructed to capture the connectivity between the harvested terms. A *Hyponym Pattern Linkage Graph (HPLG)* is created, which is defined as a graph $G = (V, E)$, where each vertex $v \in V$ is a candidate term and each edge $(u, v) \in E$ means that term u generated term v. The weight of the edge is the frequency with which u generated v. The concepts can then ranked by their *productivity*, which is represented as the out-degree of each node (term) in the graph. The out-degree of vertex v is *outD(v)* = $\frac{\sum_{\forall(v,p)\in E} w(v,p)}{|V|-1}$. Terms with out-degree > 0 are considered to be true ("trusted") hyponym terms. The output of the hyponym bootstrapping algorithm is a list of hyponym terms ranked by their out-degree score.

Bootstrapped Learning of Hypernym and Hyponym Terms

In this section, we present a new bootstrapping algorithm that harvests the web for both hypernyms and hyponyms. As before, we begin with two seed words, one hypernym (the Root concept) and one hyponym, which are instantiated in the doubly anchored hyponym pattern. The goal of our new bootstrapping algorithm, however, is not just to learn additional hyponyms of the Root concept but also to learn additional hypernyms. For example, suppose that the Root concept is *animal* and the seed hyponym is *lion*. We want to learn additional superordinate category terms that are also hypernyms of the word *lion*, such as *mammal* and *predator*. Our new algorithm consists of two internal modules that alternately learn hyponyms and hypernyms. The purpose of jointly learning hypernyms and hyponyms is twofold: (1) we hope to thoroughly explore the concept space underneath the Root concept, to learn terms that correspond to intermediate concepts that ultimately could be used to structure a taxonomy, and (2) acquiring new hypernym terms allows us to re-instantiate the doubly-anchored hyponym pattern with new seed hypernyms, which reinvigorates the bootstrapping process and allows more hyponym terms to be learned.

The first module consists of the hyponym bootstrapping algorithm discussed in the previous section. We made one modification to do some additional bookkeeping and keep track of the pairs of conjoined hyponyms that were discovered by the web queries. For example, consider the web query: *animals such as lions and **. If this query discovers the hyponyms *tigers* and *bears*, then we store the "hyponym pairs" *lions,tigers* and *lions,bears* in a table. These hyponym pairs will be used during hypernym learning. These pairs represent examples that people naturally joined together in their text to exemplify a concept, and so we hypothesized that they are likely to be representative of the concept.

The second module of our bootstrapping algorithm focuses on the generation of new hypernym terms. The algorithm uses the hyponym pairs collected during learning to instantiate a variant of the doubly-anchored pattern in which the hypernym position is left blank. We will call this a *doubly-anchored hypernym pattern*, which has the following form:

"* *such as* $< hyponym_1 >$ *and* $< hyponym_2 >$"

for example,

"* *such as lions and tigers*"

We instantiate this pattern with every hyponym pair that is collected during the previous bootstrapping step, issuing each instantiated pattern as a web query and extracting new hypernym terms from the position of the asterisk (*).

At the end of the hypernym acquisition process, we have a large number of candidate hypernym terms. In principle, we would like to use each one to instantiate the doubly-anchored hyponym pattern and perform additional hyponym learning. However, it is not practical to feed them all back into the hyponym bootstrapping loop because the number of extracted hypernyms is large and we have only a limited ability to issue web queries. Furthermore, not all of the learned hypernym terms are true hypernyms, so we need some way to choose the best ones.

We decided to rank the hypernym terms and identify the "best" hypernyms to use for bootstrapping. We use two selection criteria: (1) the hypernym should be prolific (i.e., produce many hyponyms) in order to keep the bootstrapping process energized, and (2) the hypernym should be subordinate to the original Root concept (i.e., the original seed hypernym that began the entire process), so that the learned concepts stay in the targeted part of the search space.

To rank the harvested hypernym terms, we created a new kind of *Hyponym Pattern Linkage Graph (HPLG)* based on the doubly-anchored hypernym pattern. We define a bipartite graph $G' = (V', E')$ that has two types of vertices. One set of vertices represents the hypernym terms that were harvested. We will call these *category vertices (V_c)*. The second set of vertices represents the hyponym pairs that produced the new hypernym terms. We will call these *member pair vertices (V_{mp})*. We create an edge $e'(u', v') \in E'$ between $u' \in V_c$ and $v' \in V_{mp}$ when the category represented by u' was harvested by the member pair represented by v', with the weight of the edge defined as the number of times that the member pair found the category (hypernym) term.

For example, imagine that the pattern “* *such as lions and tigers*” harvests the hypernym terms “*mammals*” and “*felines*”. The bipartite graph will contain two vertices u_1 and u_2 for the categories “*mammals*” and “*felines*”, respectively, and one vertex v_3 for the member pair $< lions, tigers >$, with two edges $e_1(u_1, v_3)$ and $e_2(u_2, v_3)$.

Once the bipartite graph is constructed, we can rank the hypernym terms using the popularity-based HPLG graph measure defined in (Kozareva, Riloff, & Hovy 2008). The *popularity* of vertex $u' \in V_c$ is defined as the in-degree score, which is computed as *inD*(u') = $\frac{\sum_{\forall(u',v')\in E'} w(u',v')}{|V'|-1}$, where $V' = V_c \cup V_{mp}$. Intuitively, a hypernym will be ranked highly if it was harvested by a diverse set of hyponym pairs.

The ranking criterion ranks the hypernyms, but does not differentiate them as being more or less general than the Root category. In order to prevent the algorithm from absorbing increasingly general concepts and wandering further and further from the original concept, it is necessary to constrain the search process to remain ‘below’ the Root category. For example, when harvesting animal categories, the system may learn the word “*species*”, which is a common term associated with animals, but this concept is superordinate to the term “animal” because it also applies to non-animals such as plants.

To constrain the bootstrapping process, we use a *Concept Positioning Test*. For this purpose, we instantiate the doubly-anchored hyponym pattern with the candidate hypernym and the original Root concept in two ways, as shown below:

(a) <*Hypernym*> *such as* <*Root*> *and* *
(b) <*Root*> *such as* <*Hypernym*> *and* *

If the candidate hypernym produces many web hits in query (a), then that suggests that the term is superordinate to the Root concept. For example, we would expect “*animals such as birds*” to produce more web hits than “*birds such as animals*”. Conversely, if the candidate hypernym produces many web hits in query (b), then that suggests that the term is subordinate to the Root concept.

Since web hits are a very coarse measure, our Concept Positioning Test simply checks that the candidate hypernym produces many more hits in pattern (b) than in pattern (a). Specifically, if pattern (b) returns at least four times as many web hits as pattern (a), and pattern (b) returned at least 50 hits, then the hypernym passes the test. Otherwise it fails. The requirement of 50 hits is just to ensure that the hypernym is a frequent concept, which is important for the bootstrapping process to maintain momentum. These values were chosen arbitrarily without much experimentation, so it is possible that other values could perform better.

To select the “best” hypernym to use for bootstrapping, we walk down the ranked list of hypernyms and apply the Concept Positioning Test. The first hypernym that passes the test is used for expansion in the next bootstrapping cycle.

We evaluated the performance of our bootstrapping algorithm on two categories: *animals* and *people*. We selected these categories because they have a large conceptual space and capture the complexity of the task.

Table 1 shows the 10 top-ranked hyponyms and hypernyms for the animal and people categories. The hyponym concepts, denoted as AHypo for animals and PHypo for people, were ranked with the HPLG outdegree measure. The hypernym concepts, denoted as AHyper for animals and PHyper for people, were ranked based on the indegree measure and the concept positioning test.

#Ex.	AHypo	AHyper	PHypo	PHyper
1	dogs	insect	Jesse Jackson	leader
2	kudu	bird	Paris Hilton	reformer
3	cats	specie	Bill Clinton	celebrity
4	sheep	invertebrate	Bill Gates	prophet
5	rats	predator	Brad Pitt	artist
6	mice	mammal	Moses	star
7	rabbits	pest	Tiger Woods	dictator
8	horses	pet	Gandhi	writer
9	pigs	crustacean	Donald Trump	teacher
10	cows	herbivore	Picasso	poet

Table 1: *Top 10 harvested concepts*

Taxonomization Framework

In order to evaluate the results, we initially considered manually defining animal and people hierarchy structures, but did not have a good sense of what the structure should be or how rich the space might get. As soon as we began to see the hypernyms that were being learned, we realized that the concept space was even more diverse and complex than we had originally anticipated.

We learned a stunningly diverse set of hypernym terms (subconcepts). Some of the learned animals terms were the expected types of concepts, such as *mammals*, *pets*, and *predators*. Even when just considering these concepts, it became clear that the taxonomic structure must allow for a word to have multiple hypernyms (e.g., a cat is simultaneously a mammal, a pet, and a predator). We also learned many highly specific animal subconcepts, such as *laboratory animals*, *forest dwellers*, and *endangered species*, as well as slang-ish animal terms, such as *pests* and *vermin*. Many of the learned terms were food words, some of which seem ambiguous as to whether they refer to the animal itself or a food product (e.g., *seafood* or *poultry*), while other terms more clearly describe just a food product (e.g., *beef* or *meat*). Another complication was that some of the learned terms were relative concepts that are hard to define in an absolute sense, such as *native animals* (native to where?) and *large mammals* (is there a general consensus on which mammals are large?).

Given the rather daunting diversity of learned terms, we decided to embark on an extensive human annotation effort to assess the nature of the categories and to find out whether human annotators would consistently agree on these concepts. Our first step was to create an extensive set of annotation guidelines. This effort itself was a valuable exercise, requiring us to think hard about the different types of concepts that exist and how we might distinguish them in a meaningful yet general way. For instance, animals come in different shapes and sizes, they inhabit land, air and water. Some of the terms associated with animals represent their feeding habits (e.g., *grazers*), the shape of the teeth (e.g., sharp for eating meat and flat for grinding and chewing plants). Other concepts relate to the adaptations of the animals for protection, for movement and for caring for their young among others. The richness of the domain predisposes some of the concepts to play the role of bridges between the super and subordinate concepts.

In the following subsections, we describe the classes that we defined in the annotation guidelines for animal and people terms. For each class, we provide a definition and some examples of terms that belong to the class. These annotation guidelines represent our first attempt at a preliminary taxonomic framework.

Animal annotation guidelines

For animal concepts, we defined fourteen classes:

1. BasicAnimal – The **basic individual** animal. Can be visualized mentally. Examples: Dog, Snake, Hummingbird.
2. GeneticAnimalClass – A **group** of basic animals, defined by **genetic similarity**. Cannot be visualized as a specific type. Examples: Reptile, Mammal. Sometimes a genetic class is also characterized by distinctive behavior, and so should be coded twice, as in Sea-mammal being both GeneticAnimalClass and BehavioralByHabitat.
3. NonRealAnimal – **Imaginary** animals. Examples: Dragon, Unicorn.
4. BehavioralByFeeding – A type of animal whose essential defining characteristic relates to a **feeding pattern** (either feeding itself, as for Predator or Grazer, or of another feeding on it, as for Prey). Cannot be visualized as an individual animal.
5. BehavioralByHabitat – A type of animal whose essential defining characteristic relates to its habitual or otherwise noteworthy **spatial location**. Cannot be visualized as an individual animal. Examples: Saltwater mammal, Desert animal.
6. BehavioralBySocializationIndividual – A type of animal whose essential defining characteristic relates to its patterns of **interaction with other animals**, of the same or a different kind. Excludes patterns of feeding. May be visualized as an individual animal. Examples: Herding animal, Lone wolf.
7. BehavioralBySocializationGroup – A natural **group of basic** animals, defined by **interaction with other animals**. Cannot be visualized as an individual animal. Examples: Herd, Pack.
8. MorphologicalTypeAnimal – A type of animal whose essential defining characteristic relates to its internal or external **physical structure** or appearance. Cannot be visualized as an individual animal. Examples: Cloven-hoofed animal, Short-hair breed.
9. RoleOrFunctionOfAnimal – A type of animal whose essential defining characteristic relates to the **role or function** it plays with respect to others, typically humans. Cannot be visualized as an individual animal. Examples: Zoo animal, Pet, Parasite, Host.
10. GeneralTerm – A term that includes animals (or humans) but **refers also to things that are neither animal nor human**. Typically either a very general word such as Individual or Living being, or a general role or function such as Model or Catalyst.
11. EvaluativeTerm – A term for an animal that carries an **opinion judgment**, such as “varmint”. Sometimes a term has two senses, one of which is just the animal, and the other is a human plus a connotation. For example, “snake” or “weasel” is either the animal proper or a human who is sneaky.
12. OtherAnimal – Almost certainly an animal or human, but **none of the above** applies, or: “I simply don’t know enough about the animal to know where to classify it”.
13. NotAnimal – **Not an animal or human**, but a real English term nonetheless.
14. GarbageTerm – **Not a real English** word.

People annotation guidelines

For people concepts, we defined the following classes:

1. BasicPerson – The **basic individual** person or persons. Can be visualized mentally. Examples: Child, Woman.
2. GeneticPersonClass – A person or persons defined by **genetic characteristics/similarity**. Can be visualized as a specific type. Examples: Asian, Saxon.
3. ImaginaryPeople – **Imaginary individuals or groups**. Examples: Superman, the Hobbits.
4. RealPeople – **Specific real individuals or groups**, by name or description. Example: Madonna, Mother Theresa, the Beatles.
5. NonTransientEventParticipant – The **role a person plays consistently over time**, by taking part in one or more specific well-defined events. This class distinguishes from

PersonState, since there is always an associated characteristic **action or activity that either persists or recurs**, without a specific endpoint being defined. This group includes several types, including: Occupations (priest, doctor), Hobbies (skier, collector), Habits (stutterer, peacemaker, gourmand).

6. TransientEventParticipant – The **role a person plays for a limited time**, through taking part in one or more specific well-defined events. There is always an associated characteristic **action or activity**, with a defined (though possibly unknown) **endpoint**. The duration of the event is typically from hours to days, perhaps up to a year, but certainly less than a decade. Examples: speaker, passenger, visitor.
7. PersonState – A person with a **certain physical or mental characteristic that persists over time**. Distinguishing this class from NonTransientEventParticipant, there is **no typical associated defining action** or activity that one can think of. Examples: midget, schizophrenic, AIDS patient, blind person.
8. FamilyRelation – A **family relation**. Examples: aunt, mother. This is a specialized subcategory of SocialRole, so don't code family relations twice.
9. SocialRole – The **role a person plays in society**. Unlike NonTransientEventParticipant, there is no single associated defining event or activity, but rather a collection of possible ones together. Examples: role model, fugitive, alumnus, hero, star, guest.
10. NationOrTribe – A nationality or tribal affiliation. Examples: Bulgarian, American, Swiss, Zulu. "Aboriginal" is a GeneticPersonClass, not a NationOrTribe.
11. ReligiousAffiliation – A **religious affiliation**. Examples: Catholic, atheist. Some religious affiliations, notably being Jewish, have strong NationOrTribe connotations as well, therefore both labels should be coded for such term.
12. OtherHuman – Clearly a human and not an animal or other being, but **does not fit into any other class**.
13. GeneralTerm – Can be a human, but **also includes other non-human entities**. Examples: image, example, figure.
14. NotPerson – Simply **not a person**.

Looking at the examples in the class definitions, it can be seen that the taxonomization task is challenging, because a term can belong to multiple categories at the same time. In addition, the taxonomy structure can have multiple facets. The initial goal of our research is to partition the learned terms into these thematically related groups.

Taxonomization Tests

We assessed the performance of our bootstrapping algorithm both on hyponym learning and hypernym learning, in separate evaluations.

Hyponym Learning Evaluation

To evaluate the quality of the learned hyponym terms for the animal and people concepts, we employed two different evaluation methods. Typically the animal terms are common nouns such as *dog* and *duck*. We compiled a gold standard list of animal terms and conducted an automatic evaluation for the animal category.[1] For people, however, the hyponym terms were primarily proper names, such as *Madonna*, *David Beckham* and *David Jones*. It is difficult to find a comprehensive list of people names, so we conducted a manual evaluation of the people terms. We randomly selected 200 people terms and asked three human annotators to tag each term as "Person" if it is a person name, and "Not-Person" otherwise. During annotation, the annotators were encouraged to consult external resources such as the World Wide Web and Wikipedia.

We ran the bootstrapping algorithm for ten iterations, both for the animal and people categories. Table 2 shows the accuracy of the learned animal hyponyms after each iteration. The accuracy is calculated as the percentage of the learned terms that are found in the gold standard list. The second raw of Table 2 shows the number of hyponyms that the algorithm has learned after each iteration. Note that the first iteration corresponds to the results of our original hyponym bootstrapping, before any hypernyms have been learned. Subsequent iterations show the number of additional hyponym terms that are learned through the bootstrapping of the learned hypernyms.

It.	1	2	3	4	5	6	7	8	9	10
Acc	.79	.79	.78	.70	.68	.68	.67	.67	.68	.71
# Hyp	396	448	453	592	663	708	745	755	770	913

Table 2: *Animal Hyponym Term Evaluation*

During the early stages of bootstrapping, the accuracy is close to 80%. As bootstrapping progresses, accuracy decreases and levels off at about 70%. The algorithm harvested 913 unique animal terms after 10 iterations. It should be noted that our animal gold standard is still far from complete, so our accuracy results are conservative estimates of the true accuracy. It is nearly impossible to create a truly comprehensive list of animal terms for several reasons, such as multiple common names (e.g.,"cougar", "panther" and "mountain lion" all refer to the same animal), spelling variants (e.g., "hyena", "hyaena") and slang short-hand terms (e.g. "hippo", "hippopotomus"; "rhino", "rhinocerus").

For the people category, the term learning algorithm generated 1549 unique names in 10 bootstrapping iterations. Table 3 shows the annotation results of the three annotators for

[1] We identified 3 web sites that catalog photographs or drawings of animals and compiled an extensive list of animal terms from their indices. These web sites are birdguides.com/species/default.asp?list=11&menu=menu_species, www.lib.ncsu.edu/findingaids/xml/mc00285.xml#id1524944, and calphotos.berkeley.edu/fauna/. To further improve the coverage of our gold standard, we also added the leaf node terms (i.e., category members) from the San Diego Zoo's animal thesaurus (http://library.sandiegozoo.org/thesaurus.htm), and animal lists acquired from Wikipedia. In total, our gold standard list of animal members contains 3939 terms. Since we only generate 1-word hyponyms, we compared against only the head nouns of the gold standard items (e.g., "blue jay" was reduced to "jay").

the 200 randomly selected terms. The average accuracy of the annotators is **0.95%**.

code	A_A	A_B	A_C
Person	190	192	189
NotPerson	10	8	11
Accuracy	.95	.96	.95

Table 3: *People Hyponym Term Evaluation*

We conclude that our bootstrapping algorithm is able to learn large quantities of high-quality hyponym terms associated with a Root concept (the seed hypernym). Table 2 demonstrates that many hyponyms are learned as a results of learning new hypernyms that are then bootstrapped into the hyponym learning process. As a reminder, all of these terms are learned using only one seed hypernym and one seed hyponym as input.

Hypernym Term Learning Evaluation

As we explained earlier, the evaluation of the learned hypernyms is more difficult because the algorithm learns a tremendously diverse set of concept terms. Ideally, we would like to obtain the internal category structure between the input Root concept and the subconcept, in order to begin building up term networks and ontologies. Unfortunately, by harvesting real text, the algorithm learns many more, and more different, terms than the ones typically contained in the neat tree-like hierarchies often shown in term taxonomies and ontologies. A dog may be a Pet, an Animal, a Carnivore, a Hunter, a Mammal, and a Performing Animal simultaneously, and these concepts do not fall into a simple tree structure. We are currently investigating automated methods to identify groupings of these concepts into ontologically parallel families, such as Predator/Prey and Carnivore/Herbivore/Omnivore. The problem is to determine what families there can be.

In the present work, we treat the hypernym/hyponym relation as ambiguous between the mathematical operators subset-of ($\subseteq$) and element-of ($\in$), and accept as correct any concept (set) to which the subordinate concept (or subset) may belong. This allows us to treat Dog and Cat as hyponyms for animals, and Madonna and Ghandi as hyponyms for people, even though strictly speaking they are of different ontological types: Dogs and Cats are sets of individuals while Madonna and Ghandi are individuals.

We do not want to preclude hypernymy relationships that are not present in resources such as WordNet (Fellbaum 1998) and CYC[2], but that are correct according to this expanded view, so we consider simply whether a term can be a hyponym of another term. To establish a gold standard, we asked four independent annotators (two graduate students and two undergrads, all native English speakers, all experienced annotators employed at a different institution) to assign each learned hypernym term to one or more of the classes defined in the annotation guidelines shown earlier. Using the Coding Analysis Toolkit (CAT)[3], the annotators were presented with the term plus three sentence-length contexts from which the term was extracted. They were encouraged to employ web search, notably Wikipedia, to determine the meanings of a term.

Table 4 shows the results from the classifications of the learned hypernyms for the animal category. In total the annotators classified 437 animal terms into the fourteen classes we defined. The first column of the table denotes the code of the class label. Columns 2 to 5 correspond to the number of times a label was assigned by an annotator. Column "Ex" denotes the number of exact matches for the class of a term between all four annotators. Column "Pa" denotes the number of partial matches for the class of a term. The final column shows the Kappa agreement of the four annotators for a class. In the current implementation of the Kappa measure, the CAT system considers the exact and partial matches between the four annotators.

Animal							
code	A_D	A_E	A_F	A_G	Ex	Pa	K
BasicAnimal	29	24	13	4	2	12	0.51
BehByFeeding	48	33	45	49	27	17	0.68
BehlByHabitat	85	58	56	54	36	36	0.66
BehBySocialGroup	1	2	6	7	0	3	0.47
BehBySocialInd	5	4	1	0	0	2	0.46
EvaluativeTerm	41	14	10	29	6	19	0.51
GarbageTerm	21	12	15	16	12	3	0.74
GeneralTerm	83	72	64	79	19	72	0.52
GeneticAnimalClass	95	113	81	73	42	65	0.61
MorphTypeAnimal	29	33	42	39	13	26	0.58
NonRealAnimal	0	1	0	0	0	0	0.50
NotAnimal	81	97	82	85	53	40	0.68
OtherAnimal	34	41	20	6	1	24	0.47
RoleOrFuncOfAnimal	89	74	76	47	28	56	0.58
Totals	641	578	511	488	239	375	0.57

Table 4: *Animal Hypernym Term Evaluation*

For instance, the code "BasicAnimal" was assigned to 29 hypernyms out of the 437 learned hypernyms by the first annotator. The second annotator labeled only 24 hypernyms as "BasicAnimal" terms, while the third and the fourth annotators assigned the "BasciAnimal" code to 13 and 4 hypernyms, respectively. There are only two hypernym terms on which all annotators agreed that the only label for the term is "BasicAnimal". The Kappa agreement for the assignment of the "BasicAnimal" class is 0.51.

At the bottom, the Totals row indicates the total number of classifications assigned by each annotator (remember that more than one class label can be assigned to a hypernym). The first two annotators A_D,A_E were more liberal and assigned many more labels in comparison to the third and fourth annotators. For instance, A_D assigned 641 classes to the 437 learned hypernyms. In comparison A_G assigned only 488 labels. This shows that A_G rarely assigned more than one class to a term. The majority of the learned animal hypernyms were assigned to the GeneticAnimalClass, BehaviourByHabitat, and RoleOrFunctionOfAnimal classes.

[2] www.cyc.com/

[3] http://cat.ucsur.pitt.edu/default.aspx

Table 5 shows the annotation results for the learned people hypernyms. The annotators manually classified 296 hypernyms into the fourteen people classes that we defined in the annotation guidelines. The structure and the organization of the table for people is the same as the one for animals. We can observe that also for the people category, the first two annotators assigned many more class labels in comparison to the third and fourth annotators. The majority of the learned people hypernyms relate to the TransientEvent-Participant, NonTransientEventParticipant, SocialRole, PersonState classes.

People							
code	A_D	A_E	A_F	A_G	Ex	Pa	K
BasicPerson	10	13	10	12	6	5	0.63
FamilyRelation	3	3	4	10	3	1	0.63
GeneralTerm	44	22	34	11	7	25	0.51
GeneticPersonClass	6	13	0	2	0	6	0.44
ImaginaryPeople	10	6	1	0	0	4	0.46
NationOrTribe	2	1	1	0	0	1	0.50
NonTransEvParti	101	154	126	99	75	59	0.69
NotPerson	44	35	26	46	22	21	0.68
OtherHuman	24	13	6	4	1	7	0.49
PersonState	45	0	34	3	0	22	0.44
RealPeople	2	0	0	0	0	0	0.50
ReligiousAffil	14	6	14	8	3	8	0.55
SocialRole	144	144	68	111	42	102	0.56
TransEvParti	108	6	15	16	0	20	0.48
Totals	557	416	339	322	159	281	0.54

Table 5: *People Hypernym Term Annotation*

The manual annotations reveal two things. First, the bootstrapping algorithm learns some terms that are not desirable (e.g., GarbageTerm, NotAnimal, NotPerson, GeneralTem classes). This shows that there is room for improvement to filter and remove unrelated and overly general terms with respect to the Root concept. Second, the inter-annotator agreements are mixed, with some classes getting relatively good agreement (say, above .65) but other classes getting weak agreement from the annotators. Clearly one of the biggest problems resulted from allowing multiple labels to be assigned to a term.

Related Work

Many Natural Language Processing applications utilize ontological knowledge from resources like WordNet[4], CYC, SUMO[5] among others. These knowledge repositories are high quality because they are manually created. However, they are costly to assemble and maintain as human effort is needed to keep them up to date. The trade-off of manually created resources is between high quality and low coverage. For example, often such resources will not include the latest best selling book or the Football Player of the Year.

Recent attempts to automatically learn bits of information necessary for LbR focus on concept harvesting (Pantel, Ravichandran, & Hovy 2004), (Pantel & Ravichandran 2004); relation learning (Berland & Charniak 1999; Girju, Badulescu, & Moldovan 2003), (Pantel & Pennacchiotti 2006), (Davidov, Rappoport, & Koppel 2007); or a combination of the two. Some systems take as input preclassified documents (Riloff 1996) or labeled document segments (Craven *et al.* 2000) and automatically learn domain-specific patterns. Others like DIPRE (Brin 1998) and Snowball (Agichtein & Gravano 2000) require a small set of labeled instances or a few hand-crafted patterns to launch the extraction process. Different approaches target different types of information sources. For instance, Yago (Suchanek, Kasneci, & Weikum 2007) extracts concepts and relations from Wikipedia, while (Pasca 2004), (Etzioni *et al.* 2005) and (Banko *et al.* 2007) mine the Web. Researchers have also worked on ontology discovery (Buitelaar, Handschuh, & Magnini 2004), (Cimiano & Volker 2005) and knowledge integration (Murray & Porter 1989), (Barker *et al.* 2007) algorithms.

Among the biggest automatically created ontologies is Yago (Suchanek, Kasneci, & Weikum 2007). It is built on entities and relations extracted from Wikipedia. The information is unified with WordNet using a carefully designed combination of rule-based and heuristic methods. The coverage of Yago depends on the number of entries in Wikipedia and the tagging of these entries with Wikipedia categories. A challenge which remains for Yago is the unification process of concepts for which there is no taxonomy (for instance, a taxonomy of emotions).

In comparison with Yago, the knowledge harvesting algorithm we have presented in this paper does not use any information about the organization of the concepts or the categories they can be related to. Our algorithm rather mines the Web to extract and rank the relevant from non-relevant information. Similar Web-based knowledge harvesting approach is that of DIPRE (Brin 1998). Given two seed concepts in a relationship, DIPRE identifies Web pages containing the seeds and learns the contexts in which the concepts are seen together. The algorithm extracts regular expressions from the contexts and applies them for the identification of new concepts that express the same relation.

Our work is most closely related and inspired by Hearst's (Hearst 1992) early work on hyponym learning. Hearst's system exploits patterns that explicitly identify a hyponym relation between a concept and an instance (e.g., *"such authors as Shakespeare"*). We have further exploited the power of the hyponym patterns by proposing doubly-anchored hyponym and hypernym patterns that can learn both new instances of a concept (hyponyms) and new category terms (hypernyms). We also use the hyponym pattern in a Concept Positioning Test to assess the relative position of a term with respect to a Root Concept.

(Pasca 2004) also exploits Hearst's hyponym patterns in lexico-syntactic structures to learn semantic class instances, and inserts the extracted instances into existing hierarchies such as WordNet. Other systems like KnowItAll (Etzioni *et al.* 2005) integrate Hearst's hyponym patterns to extract and compile instances of a given set of unary and binary predicate instances, on a very large scale. KnowItAll's learning process is initiated from generic templates that harvest candidate instances. The learned instances are ranked with mu-

[4] www.wordnet.princeton.edu

[5] www.ontologyportal.org

tual information and kept when the frequency is high. To improve recall, KnowItAll uses multiple seed patterns of semantically related concepts. For instance, to gain higher accuracy for the instances belonging to the concept *cities*, KnowItAll uses patterns of the type *"cities such as *"* and *"towns such as *"*.

Conclusion

We have presented a novel weakly supervised method for reading Web text, learning taxonomy terms, and identifying hypernym/hyponym relations. The bootstrapping algorithm requires minimal knowledge: just one seed hypernym and one seed hyponym as input. The core idea behind our approach is to exploit a doubly-anchored hyponym/hypernym pattern of the form: *"<hypernym> such as* $< hyponym_1 >$ *and* $< hyponym_2 >$*"*, which is instantiated in several different ways to learn both hyponyms, hypernyms, and the relative position of different terms.

Our approach offers the possibility of automatically generating term taxonomies in the future, surmounting the need for man-made resources. Our evaluation shows that our algorithm learns an extensive and high quality list of hyponym terms. The learned hypernym terms, however, were remarkably diverse and will be a challenge to classify and organize automatically. We created detailed annotation guidelines to characterize different types of conceptual classes that a term can belong to, but our inter-annotator agreements were mixed and showed that people have difficulty classifying concepts as well. In future work, we plan to further investigate these issues and methods for automatically inducing structure among the hypernym terms.

Acknowledgments

This research was supported in part by grants from the Department of Homeland Security, ONR Grant numbers N0014-07-1-0152 and N00014-07-1-0149. We are also very grateful to the annotators at the University of Pittsburgh who helped us evaluate this work: Jay Fischer, David Halpern, Amir Hussain, and Taichi Nakatani.

References

Agichtein, E., and Gravano, L. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth (ACL) International Conference on Digital Libraries*.

Banko, M.; Cafarella, M.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction from the web. In *Proceedings of International Joint Conference on Artificial Itelligence*, 2670–2676.

Barker, K.; Agashe, B.; Chaw, S. Y.; Fan, J.; Friedland, N. S.; Glass, M.; Hobbs, J. R.; Hovy, E. H.; Israel, D. J.; Kim, D. S.; Mulkar-Mehta, R.; Patwardhan, S.; Porter, B. W.; Tecuci, D.; and Yeh, P. Z. 2007. Learning by reading: A prototype system, performance baseline and lessons learned. In *Proceedings of AAAI*, 280–286.

Berland, M., and Charniak, E. 1999. Finding Parts in Very Large Corpora. In *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics*.

Brin, S. 1998. Extracting patterns and relations from world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, 172–183.

Buitelaar, P.; Handschuh, S.; and Magnini, B. 2004. Towards evaluation of text-based methods in semantic web and knowledge discovery life cycle. In *A Workshop at the 16th European Conference on Artificial Intelligence*.

Cimiano, P., and Volker, J. 2005. Towards large-scale, open-domain and ontology-based named entity classification. In *Proceeding of RANLP-05*, 166–172.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence* 188(1-2):69–113.

Davidov, D.; Rappoport, A.; and Koppel, M. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proceedings of ACL*.

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

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*.

Girju, R.; Badulescu, A.; and Moldovan, D. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *HLT-NAACL*.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*, 539–545.

Kozareva, Z.; Riloff, E.; and Hovy, E. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, 1048–1056. Association for Computational Linguistics.

Murray, K. S., and Porter, B. W. 1989. Controlling search for the consequences of new information during knowledge integration. In *Proceedings of the sixth international workshop on Machine learning*, 290–295.

Pantel, P., and Pennacchiotti, M. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of ACL*.

Pantel, P., and Ravichandran, D. 2004. Automatically labeling semantic classes. In *HLT-NAACL*, 321–328.

Pantel, P.; Ravichandran, D.; and Hovy, E. 2004. Towards terascale knowledge acquisition. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, 771.

Pasca, M. 2004. Acquisition of categorized named entities for web search. In *Proceedings of CIKM*, 137–145.

Riloff, E. 1996. Automatically constructing extraction patterns from untagged text. In *Proceedings of American Association of Artificial Intelligence, AAAI*, 41–47.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A core of semantic knowledge. In *16th International Wordl Wide Web Conference (WWW-2007)*.