A Semi-Supervised Method to Learn and Construct Taxonomies using the Web

Zornitsa Kozareva and Eduard Hovy
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
{kozareva,hovy}@isi.edu

Abstract

Although many algorithms have been developed to harvest lexical resources, few organize the mined terms into taxonomies. We propose (1) a semi-supervised algorithm that uses a root concept, a basic level concept, and recursive surface patterns to learn automatically from the Web hyponym-hypernym pairs subordinated to the root; (2) a Web based concept positioning procedure to validate the learned pairs' is-a relations; and (3) a graph algorithm that derives from scratch the integrated taxonomy structure of all the terms. Comparing results with WordNet, we find that the algorithm misses some concepts and links, but also that it discovers many additional ones lacking in WordNet. We evaluate the taxonomization power of our method on reconstructing parts of the WordNet taxonomy. Experiments show that starting from scratch, the algorithm can reconstruct 62% of the WordNet taxonomy for the regions tested.

1 Introduction

A variety of NLP tasks, including inference, textual entailment (Glickman et al., 2005; Szpektor et al., 2008), and question answering (Moldovan et al., 1999), rely on semantic knowledge derived from term taxonomies and thesauri such as WordNet. However, the coverage of WordNet is still limited in many regions (even well-studied ones such as the concepts and instances below Animals and People), as noted by researchers such as (Pennacchiotti and Pantel, 2006) and (Hovy et al., 2009) who perform automated semantic class learning. This happens because WordNet and most other existing taxonomies are manually created, which makes them difficult to maintain in rapidly changing domains, and (in the face of taxonomic complexity) makes them hard to build with consistency. To surmount these problems, it would be advantageous to have an automatic procedure that can not only augment existing resources but can also produce taxonomies for existing and new domains and tasks starting from scratch.

The main stages of automatic taxonomy induction are term extraction and term organization. In recent years there has been a substantial amount of work on term extraction, including semantic class learning (Hearst, 1992; Riloff and Shepherd, 1997; Etzioni et al., 2005; Pasca, 2004; Kozareva et al., 2008), relation acquisition between entities (Girju et al., 2003; Pantel and Pennacchiotti, 2006; Davidov et al., 2007), and creation of concept lists (Katz and Lin, 2003). Various attempts have been made to learn the taxonomic organization of concepts (Widdows, 2003; Snow et al., 2006; Yang and Callan, 2009). Among the most common is to start with a good ontology and then to try to position the missing concepts into it. (Snow et al., 2006) maximize the conditional probability of hyponym-hypernym relations given certain evidence, while (Yang and Callan, 2009) combines heterogenous features like context, co-occurrence, and surface patterns to produce a more-inclusive inclusion ranking formula. The obtained results are promising, but the problem of how to organize the gathered knowledge when there is no initial taxonomy, or when the initial taxonomy is grossly impoverished, still remains.

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1110–1118,
MIT, Massachusetts, USA, 9-11 October 2010.

The major problem in performing taxonomy construction from scratch is that overall concept positioning is not trivial. It is difficult to discover whether concepts are unrelated, subordinated, or parallel to each other. In this paper, we address the following question: *How can one induce the taxonomic organization of concepts in a given domain starting from scratch?*

The contributions of this paper are as follows:

- An automatic procedure for harvesting hyponym-hypernym pairs given a domain of interest.
- A ranking mechanism for validating the learned is-a relations between the pairs.
- A graph-based approach for inducing the taxonomic organization of the harvested terms starting from scratch.
- An experiment on reconstructing WordNet's taxonomy for given domains.

Before focusing on the harvesting and taxonomy induction algorithms, we are going to describe some basic terminology following (Hovy et al., 2009). A **term** is an English word (for our current purposes, a noun or a proper name). A **concept** is an item in the classification taxonomy we are building. A **root concept** is a fairly general concept which is located on the high level of the taxonomy. A **basic-level concept** corresponds to the Basic Level categories defined in Prototype Theory in Psychology (Rosch, 1978). For example, a *dog*, not a mammal or a collie. An **instance** is an item in the classification taxonomy that is more specific than a concept. For example, *Lassie*, not a dog or collie .

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the taxonomization framework. Section 4 discusses the experiments. We conclude in Section 5.

2 Related Work

The first stage of automatic taxonomy induction, term and relation extraction, is relatively well-understood. Methods have matured to the point of achieving high accuracy (Girju et al., 2003; Pantel and Pennacchiotti, 2006; Kozareva et al., 2008). The produced output typically contains flat lists of terms and/or ground instance facts ($lion$ is-a $mammal$) and general relation types ($mammal$ is-a $animal$).

Most approaches use either clustering or patterns to mine knowledge from structured and unstructured text. Clustering approaches (Lin, 1998; Lin and Pantel, 2002; Davidov and Rappoport, 2006) are fully unsupervised and discover relations that are not directly expressed in text. Their main drawback is that they may or may not produce the term types and granularities useful to the user. In contrast, pattern-based approaches harvest information with high accuracy, but they require a set of seeds and surface patterns to initiate the learning process. These methods are successfully used to collect semantic lexicons (Riloff and Shepherd, 1997; Etzioni et al., 2005; Pasca, 2004; Kozareva et al., 2008), encyclopedic knowledge (Suchanek et al., 2007), concept lists (Katz and Lin, 2003), and relations between terms, such as hypernyms (Ritter et al., 2009; Hovy et al., 2009) and part-of (Girju et al., 2003; Pantel and Pennacchiotti, 2006).

However, simple term lists are not enough to solve many problems involving natural language. Terms may be augmented with information that is required for knowledge-intensive tasks such as textual entailment (Glickman et al., 2005; Szpektor et al., 2008) and question answering (Moldovan et al., 1999). To support inference, (Ritter et al., 2010) learn the selectional restrictions of semantic relations, and (Pennacchiotti and Pantel, 2006) ontologize the learned arguments using WordNet.

Taxonomizing the terms is a very powerful method to leverage added information. Subordinated terms (hyponyms) inherit information from their superordinates (hypernyms), making it unnecessary to learn all relevant information over and over for every term in the language. But despite many attempts, no 'correct' taxonomization has ever been constructed for the terms of, say, English. Typically, people build term taxonomies (and/or richer structures like ontologies) for particular purposes, using specific taxonomization criteria. Different tasks and criteria produce different taxonomies, even when using the same basic level concepts. This is because most basic level concepts admit to multiple perspectives, while each task focuses on one, or at most two, perspectives at a time. For example, a dolphin is a Mammal (and not a Fish) to a biologist, but is a Fish

(and hence not a Mammal) to a fisherman or anyone building or visiting an aquarium. More confusingly, a tiger and a puppy are both Mammals and hence belong close together in a typical taxonomy, but a tiger is a WildAnimal (in the perspective of Animal-Function) and a JungleDweller (in the perspective of Habitat), while a puppy is a Pet (as function) and a HouseAnimal (as habitat), which would place them relatively far from one another. Attempts at producing a single multi-perspective taxonomy fail due to the complexity of interaction among perspectives, and people are notoriously bad at constructing taxonomies adherent to a single perspective when given terms from multiple perspectives. This issue and the major alternative principles for taxonomization are discussed in (Hovy, 2002).

It is therefore not surprising that the second stage of automated taxonomy induction is harder to achieve. As mentioned, most attempts to learn taxonomy structures start with a reasonably complete taxonomy and then insert the newly learned terms into it, one term at a time (Widdows, 2003; Pasca, 2004; Snow et al., 2006; Yang and Callan, 2009). (Snow et al., 2006) guide the incremental approach by maximizing the conditional probability over a set of relations. (Yang and Callan, 2009) introduce a taxonomy induction framework which combines the power of surface patterns and clustering through combining numerous heterogeneous features.

Still, one would like a procedure to organize the harvested terms into a taxonomic structure starting fresh (i.e., without using an initial taxonomic structure). We propose an approach that bridges the gap between the term extraction algorithms that focus mainly on harvesting but do not taxonomize, and those that accept a new term and seek to enrich an already existing taxonomy. Our aim is to perform both stages: to extract the terms of a given domain and to induce their taxonomic organization without any initial taxonomic structure and information. This task is challenging because it is not trivial to discover both the hierarchically related and the parallel (perspectival) organizations of concepts. Achieving this goal can provide the research community with the ability to produce taxonomies for domains for which currently there are no existing or manually created ontologies.

3 Building Taxonomies from Scratch

3.1 Problem Formulation

We define our task as:

Task Definition: Given a root concept, a basic level concept or an instance, and recursive lexico-syntactic patterns, (1) harvest in bootstrapping fashion hyponyms and hypernyms subordinated to the root; (2) filter out erroneous information (extracted concepts and *isa* relations); (3) organize the harvested concepts into a taxonomy structure.

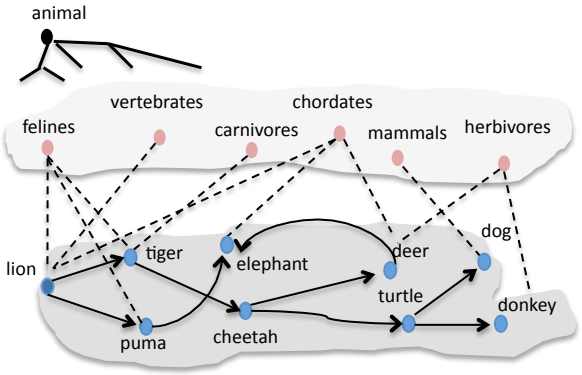

Figure 1: *Taxonomy Induction from Scratch.*

Figure 1 shows an example of the task. Starting with the root concept *animal* and the basic level concept *lion*, the algorithm learns new terms like *tiger, puma, deer, donkey* of class *animal*. Next for each basic level concept, the algorithm harvests hypernyms and learns that a *lion* is-a *vertebrate*, *chordate*, *feline* and *mammal*. Finally, the taxonomic structure of each basic level concept and its hypernyms is induced: *animal→chordate→vertebrate→mammal→feline→lion*.

3.2 Knowledge Harvesting

The main objective of our work is not the creation of a new harvesting algorithm, but rather the organization of the harvested information in a taxonomy structure starting from scratch. There are many algorithms for hyponym and hypernym harvesting from the Web. In our experiments, we use the doubly-anchored lexico-syntactic patterns and bootstrapping algorithm introduced by (Kozareva et al., 2008) and (Hovy et al., 2009).

We are interested in using this approach, because it is: (1) simple and easy to implement; (2) requires minimal supervision using only one root concept and a term to learn new hyponyms and hypernyms associated to the root; (3) reports higher precision than current semantic class algorithms (Etzioni et al., 2005; Pasca, 2004); and (4) adapts easily to different domains.

The general framework of the knowledge harvesting algorithm is shown in Figure 2.

1. Given:
   a hyponym pattern P_i={$concept$ such as $seed$ and *}
   a hypernym pattern P_c={* such as $term_1$ and $term_2$}
   a root concept $root$
   a term called $seed$ for P_i
2. build a query using P_i
3. submit P_i to Yahoo! or other search engine
4. extract terms occupying the * position
5. take terms from step 4 and go to step 2.
6. repeat steps 2–5 until no new terms are found
7. rank terms by *outDegree*
8. for $\forall$ terms with *outDegree*>0, build a query using P_c
9. submit P_c to Yahoo! or other search engine
10. extract concepts (hypernyms) occupying the * position
11. rank concepts by *inDegree*

Figure 2: *Knowledge Harvesting Framework.*

The algorithm starts with a $root$ concept, *seed* term[1] of type *root* and a doubly-anchored pattern (DAP) such as '<*root*> *such as* <*seed*> *and* *' which learns on the * position new terms of type *root*. The newly learned terms, which can be either instances, basic level or intermediate concepts, are placed into the position of the *seed* in the DAP pattern, and the bootstrapping process is repeated. The process ceases when no new terms are found.

To separate the true from incorrect terms, we use a graph-based algorithm in which each vertex u is a term, and an each edge $(u, v) \in E$ corresponds to the direction in which the term u discovered the term v. The graph is weighted $w(u, v)$ according to the number of times the term pair u-v is seen in unique web snippets. The terms are ranked by $outDegree(u)$= $\frac{\sum_{\forall(u,v)\in E} w(u,v)}{|V|-1}$ which counts the number of outgoing links of node u normalized by the total number of nodes in the graph excluding the current. The algorithm considers as true terms with $outDegree$>0.

All harvested terms are automatically fed into the hypernym extraction phase. We use the natural order in which the terms discovered each other and place them into an inverse doubly-anchored pattern (DAP^{-1}) '* *such as* <$term_1$> *and* <$term_2$>' to learn hypernyms on the * position. Similarly we build a graph with nodes h denoting the hypernyms and nodes t_1-t_2 denoting the term pairs. The edges $(h, t_1 - t_2) \in E'$ show the direction in which the term pair discovered the hypernym. The hypernyms are ranked by $inDegree(h)$= $\sum_{\forall(t_1-t_2,h)\in E'} w(t_1 - t_2, h)$ which rewards hypernyms that are frequently discovered by various term pairs. The output of the algorithm is a list of is-a relations between the learned terms (instances, basic level or intermediate concepts) and their corresponding hypernyms. For example, *deer is-a herbivore*, *deer is-a ruminant*, *deer is-a mammal*.

3.3 Graph-Based Taxonomy Induction

In the final stage of our algorithm, we induce the overall taxonomic structure using information about the pairwise positioning of the terms. In the knowledge harvesting and filtering phases, the algorithm learned is-a relations between the $root$ and the terms (instances, basic level or intermediate concepts), as well as the harvested hypernyms and the terms. The only missing information is the positioning of the intermediate concepts located between the basic level and the $root$ such as *mammals*, *vertibrates*, *felines*, *chordates*, among others.

We introduce a concept positioning (CP) procedure that uses a set of surface patterns: "X $such\ as$ Y", "X are Y $that$", "X $including$ Y", "X $like$ Y", "$such$ X as Y" to learn the hierarchical relations for all possible concept pairs. For each concept pair, say $chordates$ and $vertebrates$, we issue the two following queries:

(a) *chordates such as vertebrates*
(b) *vertebrates such as chordates*

[1]The input term can be an instance, a basic level or an intermediate concept. An intermediate concept is the one that is located between the basic level and root concepts.

If (a) returns more web hits than (b), then $chordates$ subsumes (or is broader than) $vertebrates$, otherwise $vertebrates$ subsumes $chordates$. For this pair the *such as* pattern returned 7 hits for (a) and 0 hits for (b), so that the overall magnitude of the direction of the relation is weak. To accumulate stronger evidence, we issue web queries with the remaining patterns. For the same concept pair, the overall magnitude of "X *including* Y" is 5820 hits for (a) and 0 for (b).

As shown in Figure 3, the concept positioning patterns cannot always determine the direct taxonomic organization between two concepts as in the case of *felines* and *chordates*, *felines* and *vertebrates*. One reason is that the concepts are located on distant taxonomic levels. We humans typically exemplify concepts using more proximate ones. Therefore, the concept positioning procedure can find evidence for the relation "*mammals→felines*", but not for "*chordates→felines*".

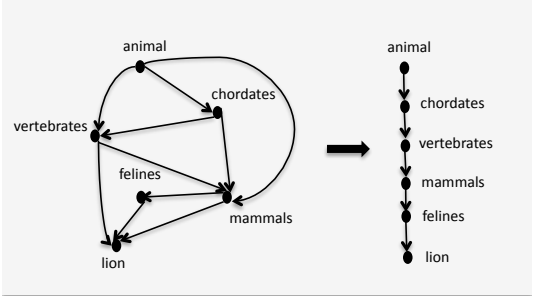

Figure 3: *Concept Positioning and Induced Taxonomy.*

After the concept positioning procedure has explored all concept pairs, we encounter two phenomena: (1) direct links between some concepts are missing and (2) multiple paths can be taken to reach from one concept to another.

To surmount these problems, we employ a graph based algorithm that finds the longest path in the graph G''=(V'', E''). The nodes V''=$\{it_1, h_1, h_2, .., h_n, r\}$ represent the input term, its hypernyms, and the $root$. An edge $(t_m, t_n) \in E''$ indicates that there is a path between the terms t_m and t_n. The direction $t_m \rightarrow t_n$ indicates the term subordination discovered during the CP procedure. The objective is to find the longest path in G'' between the $root$ and the input term. Intuitively, finding the longest paths is equivalent to finding the taxonomic organization of all concepts.

First, if present, we eliminate all cycles from the graph. Then, we find all nodes that have no predecessor and those that have no successor. Intuitively, a node with no predecessors p is likely to be positioned on the top of the taxonomy (e.g. $animal$), while a node with no successor s is likely to be located at the bottom (e.g. terms like *lion*, *tiger*, *puma*, or concepts like *krill predators* that could not be related to an instance or a basic level concept during the CP procedure). We represent the directed graph as an adjacency matrix $A = [a_{m,n}]$, where $a_{m,n}$ is 1 if (t_m, t_n) is an edge of G'', and 0 otherwise. For each (p, s) pair, we find the list of all paths connecting p with s. In the end, from all discovered candidate paths, the algorithm returns the longest one. The same graph-based taxonomization procedure is repeated for the rest of the basic level concepts and their hypernyms.

4 Experiments and Results

To evaluate the performance of a taxonomy induction algorithm, one can compare against a simple taxonomy composed of 2–3 levels. However, one cannot guarantee that the algorithm can learn larger hierarchies completely or correctly.

Animals provide a good example of the true complexity of concept organization: there are many types, they are of numerous kinds, people take numerous perspectives over them, and they are relatively well-known to human annotators. In addition, WordNet has a very rich and deep taxonomic structure for animals that can be used for direct comparison. We further evaluate our algorithm on the domains of *Plants* and *Vehicles*, which share some of these properties.

4.1 Data Collection

We have run the knowledge harvesting algorithm on the semantic classes *Animals*, *Plants* and *Vehicles* starting with only one seed example such as *lions*, *cucumbers* and *cars* respectively.

First, we formed and submitted the DAP pattern as web queries to Yahoo!Boss. We retrieved the top 1000 web snippets for each query. We kept all unique terms and term pairs. Second, we used the learned term pairs to form and submit new web

queries DAP^{-1}. In this step, the algorithm harvested the hypernyms associated with each term. We kept all unique triples composed of a hypernym and the term pairs that extracted it. The algorithm ran until complete exhaustion for 8 iterations for *Animals*, 10 iterations for *Plants* and 18 iterations of *Vehicles*.

Table 1 shows the total number of terms extracted by the Web harvesting algorithm during the first stage. In addition, we show the number of terms that passed the $outDegree$ threshold. We found that the majority of the learned terms for *Animals* are basic level concepts, while for *Plants* and *Vehicles* they are a mixture of basic level and intermediate concepts.

	Animals	Plants	Vehicles
#Extracted Terms	1855	2801	1425
#outDegree(Term)$>$ 0	858	1262	581

Table 1: *Learned Terms.*

Since human based evaluation of all harvested terms is time consuming and costly, we have selected 90 terms located at the beginning, in the middle and in the end of the $outDegree$ ranking. Table 2 summarizes the results.

Plants	#CorrectByHand	#inWN	PrecByHand
rank[1-30]	29	28	.97
rank[420-450]	29	21	.97
rank[1232-1262]	27	19	.90
Vehicles	#CorrectByHand	#inWN	PrecByHand
rank[1-30]	29	27	.97
rank[193-223]	22	18	.73
rank[551-581]	25	19	.83

Table 2: *Term Evaluation.*

Independently, we can say that the precision of the harvesting algorithm is from 73 to 90%. In the case of *Vehicles*, we found that the learned terms in the middle ranking do not refer to the meaning of vehicle as a transportation devise, but to the meaning of vehicle as media (i.e. *seminar*, *newspapers*), communication and marketing. For the same category, the algorithm learned many terms which are missing from WordNet such as *BMW*, *bakkies*, *two-wheeler*, *all-terrain-vehicle* among others.

The second stage of the harvesting algorithm concerns hypernym extraction. Table 3 shows the total number of hypernyms harvested for all term pairs. The top 20 highly ranked concepts by $inDegree$ are the most descriptive terms for the domain. However, if we are interested in learning a larger set of hypernyms, we found that $inDegree$ is not sufficient by itself. For example, highly frequent but irrelevant hypernyms such as *meats*, *others* are ranked at the top of the list, while low frequent but relevant ones such as *protochordates*, *hooved-mammals*, *homeotherms* are discarded. This shows that we need to develop additional and more sensitive measures for hypernym ranking.

	Animals	Plants	Vehicles
#Extracted Hypernyms	1904	8947	2554
#inDegree(Hypernyms)$>$ 10	110	294	100

Table 3: *Learned Hypernyms.*

Table 4 shows some examples of the learned animal hypernyms which were annotated by humans as: correct but not present in WordNet; borderline which depending on the application could be valuable to have or exclude; and incorrect.

CorrectNotInWN	{colony|social} insects, grazers, monogastrics camelid, {mammalian|land|areal} predators {australian|african} wildlife, filter feeders hard shelled invertebrates, pelagics bottom dwellers
Borderline	prehistoric animals, large herbivores pocket pets, farm raised fish, roaring cats endangered mammals, mysterious hunters top predators, modern-snakes, heavy game
Incorrect	frozen foods, native mammals, red meats furry predators, others, resources, sorts products, items, protein

Table 4: *Examples of Learned Animal Hypernyms.*

The annotators found that 9% of the harvested is-a relations are missing from WordNet. For example, *cartilaginous_fish* $\rightarrow$ *shark*; *colony_insects*$\rightarrow$ *bees*; *filter_feeders* $\rightarrow$ *tube_anemones* among others. This shows that despite its completeness, WordNet has still room for improvement.

4.2 A Test: Reconstructing WordNet

As previously discussed in (Hovy et al., 2009), it is extremely difficult even for expert to manually construct and evaluate the correctness of the harvested taxonomies. Therefore, we decided to evaluate the performance of our taxonomization approach reconstructing WordNet *Animals*, *Plants* and *Vehicles* taxonomies.

Given a domain, we select from 140 to 170 of the harvested terms. For each term, we retrieve all WordNet hypernyms located on the path between the input term and the $root$ that is *animal*, *plant* or *vehicle* depending on the domain of interest. We have found that 98% of the WordNet terms are also harvested by our knowledge acquisition algorithm. This means that being able to reconstruct WordNet's taxonomy is equivalent to evaluating the performance of our taxonomy induction approach.

Table 5 summarizes the characteristics of the taxonomies for the regions tested. For each domain, we show the total number of terms that must be organized, and the total number of is-a relations that must be induced.

	Animals	Plants	Vehicles
#terms	684	554	140
#is-a	4327	2294	412
average depth	6.23	4.12	3.91
max depth	12	8	7
min depth	1	1	1

Table 5: *Data for WordNet reconstruction.*

Among the three domains we have tested, *Animals* is the most complex and richest one. The maximum number of levels our algorithm must infer is 11, the minimum is 1 and the average taxonomic depth is 6.2. In total there are three basic level concepts (*longhorns*, *gaur* and *bullock*) with maximum depth, twenty terms (basic level and intermediate concepts) with minimum depth and ninety-eight terms (*wombat*, *viper*, *rat*, *limpkin*) with depth 6.

Plants is also a very challenging domain, because it contains a mixture of scientific and general terms such as *magnoliopsida* and *flowering plant*.

4.3 Evaluation

To evaluate the performance of our taxonomy induction approach, we use the following measures:

$$Precision = \frac{\#is{-}a\ found\ in\ WordNet\ and\ by\ system}{\#is{-}a\ found\ by\ system}$$

$$Recall = \frac{\#is{-}a\ found\ in\ WordNet\ and\ by\ system}{\#is{-}a\ found\ in\ WordNet}$$

Table 6 shows results of the taxonomy induction of the *Vehicles* domain using different concept positioning patterns. The most productive ones are: "X *are* Y *that*" and "X *including* Y". However, the highest yield is obtained when we combine evidence from all patterns.

Vehicles	Precision	Recall
X such as Y	.99 (174/175)	.42 (174/410)
X are Y that	.99 (206/208)	.50 (206/410)
X including Y	.96 (165/171)	.40 (165/410)
X like Y	.96 (137/142)	.33 (137/410)
such X as Y	.98 (44/45)	.11 (44/410)
AllPatterns	.99 (246/249)	**.60** (246/410)

Table 6: *Evaluation of the Induced Vehicle Taxonomy.*

Table 7 shows results of the taxonomization of the *Animals* and *Plants* domains. Overall, the obtained results are very encouraging given the fact that we started from scratch without the usage of any taxonomic structure. Precision is robust, but we must further improve recall. Our observation for the lower recall is that some intermediate concepts relate mostly to the high level ones, but not to the basic level concepts.

	Precision	Recall
Animals	.98 (1643/1688)	.38 (1643/4327)
Plants	.97 (905/931)	.39 (905/2294)

Table 7: *Evaluation of the Induced Animal and Plant Taxonomies.*

Figure 4 shows an example of the taxonomy induced by our algorithm for the *vipers*, *rats*, *wombats*, *ducks*, *emus*, *moths* and *penguins* basic level concepts and their WordNet hypernyms.

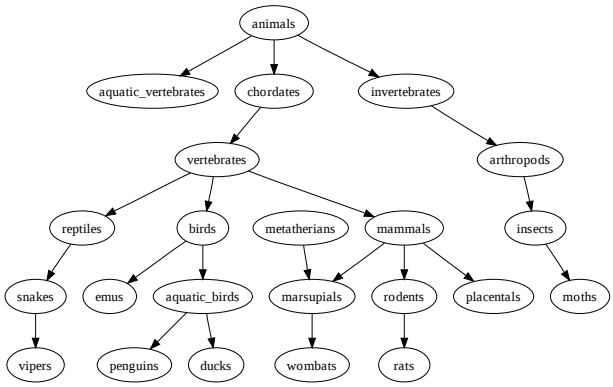

Figure 4: *Induced Taxonomy for Animals.*

The biggest challenge of the taxonomization process is the merging of independent taxonomic per-

spectives (a *deer* is a *grazer* in BehaviorByFeeding, a *wildlife* in BehaviorByHabitat, a *herd* in BehaviorSocialGroup and an *even-toed ungulate* in MorphologicalType) into a single hierarchy.

5 Conclusions and Future Work

We are encouraged by the ability of the taxonomization algorithm to reconstruct WordNet's *Animal* hierarchy, which is one of its most complete and elaborated. In addition, we have also evaluated the performance of our algorithm with the *Plant* and *Vehicle* WordNet hierarchies.

Currently, our automated taxonomization algorithm is able to build some of the quasi-independent perspectival taxonomies (Hovy et al., 2009). However, further research is required to develop methods that reliably (a) identify the number of independent perspectives a concept can take (or seems to take in the domain text), and (b) classify any harvested term into one or more of them. The result would greatly simplify the task of the taxonomization stage.

We note that despite this richness, WordNet has many concepts like *camelid*, *filter feeder*, *monogastrics* among others which are missing, but the harvesting algorithm can provide. Another promising line of research would investigate the combination of the two styles of taxonomization algorithms: first, the one described here to produce an initial (set of) taxonomies, and second, the term-insertion algorithms developed in prior work.

Acknowledgments

We acknowledge the support of DARPA contract number FA8750-09-C-3705. We thank Mark Johnson for the valuable discussions on taxonomy evaluation. We thank the reviewers for their useful feedback and suggestions.

References

Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 297–304.

Dmitry Davidov, Ari Rappoport, and Moshel Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 232–239.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165(1):91–134, June.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 1–8.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. A probabilistic classification approach for lexical textual entailment. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, pages 1050–1055.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545.

Eduard H. Hovy, Zornitsa Kozareva, and Ellen Riloff. 2009. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 948–957.

Eduard Hovy. 2002. Comparing sets of semantic relations in ontologies. *The Semantics of Relationships: An Interdisciplinary Perspective*, pages 91–110.

Boris Katz and Jimmy Lin. 2003. Selectively using relations to improve precision in question answering. In *In Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering*, pages 43–50.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056.

Dekang Lin and Patrick Pantel. 2002. Concept discovery from text. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774.

Dan I. Moldovan, Sanda M. Harabagiu, Marius Pasca, Rada Mihalcea, Richard Goodrum, Roxana Girju, and Vasile Rus. 1999. Lasso: A tool for surfing the answer net. In *TREC*.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of 21st International Conference on Computational Linguistics and*

44th Annual Meeting of the Association for Computational Linguistics, ACL 2006.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145.

Marco Pennacchiotti and Patrick Pantel. 2006. Ontologizing semantic relations. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 793–800.

Ellen Riloff and Jessica Shepherd. 1997. A Corpus-Based Approach for Building Semantic Lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124.

Alan Ritter, Stephen Soderland, and Oren Etzioni. 2009. What is this, anyway: Automatic hypernym discovery. In *Proceedings of AAAI Spring Symposium on Learning by Reading and Learning to Read*.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *to appear in Proceedings of the Association for Computational Linguistics ACL2010*.

Eleanor Rosch. 1978. Principles of categorization.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, ACL*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Idan Szpektor, Ido Dagan, Roy Bar-Haim, and Jacob Goldberger. 2008. Contextual preferences. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 683–691.

Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of HLT-NAACL*.

Hui Yang and Jamie Callan. 2009. A metric-based framework for automatic taxonomy induction. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 271–279.