Automatically Labeling Semantic Classes

Patrick Pantel and Deepak Ravichandran
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
{pantel,ravichan}@isi.edu

Abstract

Systems that automatically discover semantic classes have emerged in part to address the limitations of broad-coverage lexical resources such as WordNet and Cyc. The current state of the art discovers many semantic classes but fails to label their concepts. We propose an algorithm labeling semantic classes and for leveraging them to extract *is-a* relationships using a top-down approach.

1 Introduction

The natural language literature is rich in theories of semantics (Barwise and Perry 1985; Schank and Abelson 1977). However, WordNet (Miller 1990) and Cyc (Lenat 1995) aside, the community has had little success in actually building large semantic repositories. Such broad-coverage lexical resources are extremely useful in applications such as word sense disambiguation (Leacock, Chodorow and Miller 1998) and question answering (Pasca and Harabagiu 2001).

Current manually constructed ontologies such as WordNet and Cyc have important limitations. First, they often contain rare senses. For example, WordNet includes a rare sense of *computer* that means 'the person who computes'. Using WordNet to expand queries to an information retrieval system, the expansion of *computer* will include words like *estimator* and *reckoner*. Also, the words *dog*, *computer* and *company* all have a sense that is a hyponym of *person*. Such rare senses make it difficult for a coreference resolution system to use WordNet to enforce the constraint that personal pronouns (e.g. *he* or *she*) must refer to a person. The second problem with these lexicons is that they miss many domain specific senses. For example, WordNet misses the user-interface-object sense of the word *dialog* (as often used in software manuals). WordNet also contains a very poor coverage of proper nouns.

There is a need for (semi-) automatic approaches to building and extending ontologies as well as for validating the structure and content of existing ones. With the advent of the Web, we have access to enormous amounts of text. The future of ontology growing lies in leveraging this data by harvesting it for concepts and semantic relationships. Moreover, once such knowledge is discovered, mechanisms must be in place to enrich current ontologies with this new knowledge.

To address some of the coverage and specificity problems in WordNet and Cyc, Pantel and Lin (2002) proposed and algorithm, called CBC, for automatically extracting semantic classes. Their classes consist of clustered instances like the three shown below:

(A) multiple sclerosis, diabetes, osteoporosis, cardiovascular disease, Parkinson's, rheumatoid arthritis, heart disease, asthma, cancer, hypertension, lupus, high blood pressure, arthritis, emphysema, epilepsy, cystic fibrosis, leukemia, hemophilia, Alzheimer, myeloma, glaucoma, schizophrenia, ...

(B) Mike Richter, Tommy Salo, John Vanbiesbrouck, Curtis Joseph, Chris Osgood, Steve Shields, Tom Barrasso, Guy Hebert, Arturs Irbe, Byron Dafoe, Patrick Roy, Bill Ranford, Ed Belfour, Grant Fuhr, Dominik Hasek, Martin Brodeur, Mike Vernon, Ron Tugnutt, Sean Burke, Zach Thornton, Jocelyn Thibault, Kevin Hartman, Felix Potvin, ...

(C) pink, red, turquoise, blue, purple, green, yellow, beige, orange, taupe, white, lavender, fuchsia, brown, gray, black, mauve, royal blue, violet, chartreuse, teal, gold, burgundy, lilac, crimson, garnet, coral, grey, silver, olive green, cobalt blue, scarlet, tan, amber, ...

A limitation of these concepts is that CBC does not discover their actual names. That is, CBC discovers a semantic class of Canadian provinces such as Manitoba, Alberta, and Ontario, but stops short of labeling the concept as *Canadian Provinces*. Some applications such as question answering would benefit from class labels. For example, given the concept list (B) and a label *goalie/goaltender*, a QA system could look for answers to the question "*Which goaltender won the most Hart Trophys?*" in the concept.

In this paper, we propose an algorithm for automatically inducing names for semantic classes and for finding instance/concept (*is-a*) relationships. Using concept signatures (templates describing the prototypical syntactic behavior of instances of a concept), we extract concept names by searching for simple syntactic patterns such as "*concept apposition-of instance*". Searching concept signatures is more robust than searching the syntactic features of individual instances since many instances suffer from sparse features or multiple senses.

Once labels are assigned to concepts, we can extract a hyponym relationship between each instance of a concept and its label. For example, once our system labels list (C) as *color*, we may extract relationships such as: *pink* is a *color*, *red* is a *color*, *turquoise* is a *color*, etc. Our results show that of the 159,000 hyponyms we extract using this simple method, 68% are correct. Of the 65,000 proper name hyponyms we discover, 81.5% are correct.

The remainder of this paper is organized as follows. In the next section, we review previous algorithms for extracting semantic classes and hyponym relationships. Section 3 describes our algorithm for labeling concepts and for extracting hyponym relationships. Experimental results are presented in Section 4 and finally, we conclude with a discussion and future work.

2 Previous Work

There have been several approaches to automatically discovering lexico-semantic information from text (Hearst 1992; Riloff and Shepherd 1997; Riloff and Jones 1999; Berland and Charniak 1999; Pantel and Lin 2002; Fleischman et al. 2003; Girju et al. 2003). One approach constructs automatic thesauri by computing the similarity between words based on their distribution in a corpus (Hindle 1990; Lin 1998). The output of these programs is a ranked list of similar words to each word. For example, Lin's approach outputs the following top-20 similar words of *orange*:

(D) peach, grapefruit, yellow, lemon, pink, avocado, tangerine, banana, purple, Santa Ana, strawberry, tomato, red, pineapple, pear, Apricot, apple, green, citrus, mango

A common problem of such lists is that they do not discriminate between the senses of polysemous words. For example, in (D), the *color* and *fruit* senses of *orange* are mixed up.

Lin and Pantel (2001) proposed a clustering algorithm, UNICON, which generates similar lists but discriminates between senses of words. Later, Pantel and Lin (2002) improved the precision and recall of UNICON clusters with CBC (Clustering by Committee). Using sets of representative elements called committees, CBC discovers cluster centroids that unambiguously describe the members of a possible class. The algorithm initially discovers committees that are well scattered in the similarity space. It then proceeds by assigning elements to their most similar committees. After assigning an element to a cluster, CBC removes their overlapping features from the element before assigning it to another cluster. This allows CBC to discover the less frequent senses of a word and to avoid discovering duplicate senses.

CBC discovered both the *color* sense of *orange*, as shown in list (C) of Section 1, and the *fruit* sense shown below:

(E) peach, pear, apricot, strawberry, banana, mango, melon, apple, pineapple, cherry, plum, lemon, grapefruit, orange, berry, raspberry, blueberry, kiwi, ...

There have also been several approaches to discovering hyponym (*is-a*) relationships from text. Hearst (1992) used seven lexico-syntactic patterns, for example "*such NP as {NP,}*{(or|and)} NP*" and "*NP {, NP}*{,} or other NP*". Berland and Charniak (1999) used similar pattern-based techniques and other heuristics to extract meronymy (part-whole) relations. They reported an accuracy of about 55% precision on a corpus of 100,000 words. Girju, Badulescu and Moldovan (2003) improved upon this work by using a machine learning filter. Mann (2002) and Fleischman et al. (2003) used part of speech patterns to extract a subset of hyponym relations involing proper nouns.

3 Labeling Classes

The research discussed above on discovering hyponym relationships all take a bottom up approach. That is, they use patterns to independently discover semantic relationships of words. However, for infrequent words, these patterns do not match or, worse yet, generate incorrect relationships.

Ours is a top down approach. We make use of co-occurrence statistics of semantic classes discovered by algorithms like CBC to label their concepts. Hyponym relationships may then be extracted easily: one hyponym per instance/concept label pair. For example, if we labeled concept (A) from Section 1 with *disease*, then we could extract *is-a* relationships such as: *diabetes is a disease*, *cancer is a disease*, and *lupus is a disease*. A concept instance such as *lupus* is assigned a hypernym *disease* not because it necessarily occurs in any particular syntactic relationship with *disease*, but because it belongs to the class of instances that does.

The input to our labeling algorithm is a list of semantic classes, in the form of clusters of words, which may be generated from any source. In our experiments, we used the clustering outputs of CBC (Pantel and Lin 2002). The output of the system is a ranked list of concept names for each semantic class.

In the first phase of the algorithm, we extract feature vectors for each word that occurs in a semantic class. Phase II then uses these features to compute grammatical signatures of concepts using the CBC algorithm. Finally, we use simple syntactic patterns to discover class names from each class' signature. Below, we describe these phases in detail.

3.1 Phase I

We represent each word (concept instance) by a feature vector. Each feature corresponds to a context in which the word occurs. For example, "catch __" is a verb-object context. If the word *wave* occurred in this context, then the context is a feature of *wave*.

We first construct a frequency count vector $C(e) = (c_{e1}, c_{e2}, \ldots, c_{em})$, where m is the total number of features and c_{ef} is the frequency count of feature f occurring in word e. Here, c_{ef} is the number of times word e occurred in a grammatical context f. For example, if the word *wave* occurred 217 times as the object of the verb *catch*, then the feature vector for *wave* will have value 217 for its "*object-of catch*" feature. In Section 4.1, we describe how we obtain these features.

We then construct a mutual information vector $MI(e) = (mi_{e1}, mi_{e2}, \ldots, mi_{em})$ for each word e, where mi_{ef} is the pointwise mutual information between word e and feature f, which is defined as:

$$mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\frac{\sum_{i=1}^{n} c_{if}}{N} \times \frac{\sum_{j=1}^{m} c_{ej}}{N}} \qquad (1)$$

where n is the number of words and $N = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}$ is the total frequency count of all features of all words.

Mutual information is commonly used to measure the association strength between two words (Church and Hanks 1989). A well-known problem is that mutual information is biased towards infrequent elements/features. We therefore multiply mi_{ef} with the following discounting factor:

$$\frac{c_{ef}}{c_{ef}+1} \times \frac{\min\left(\sum_{i=1}^{n} c_{ei}, \sum_{j=1}^{m} c_{jf}\right)}{\min\left(\sum_{i=1}^{n} c_{ei}, \sum_{j=1}^{m} c_{jf}\right)+1} \qquad (2)$$

3.2 Phase II

Following (Pantel and Lin 2002), we construct a committee for each semantic class. A committee is a set of representative elements that unambiguously describe the members of a possible class.

For each class c, we construct a matrix containing the similarity between each pair of words e_i and e_j in c using the cosine coefficient of their mutual information vectors (Salton and McGill 1983):

$$sim(e_i, e_j) = \frac{\sum_f mi_{e_i f} \times mi_{e_j f}}{\sqrt{\sum_f mi_{e_i f}^{2} \times \sum_f mi_{e_j f}^{2}}} \qquad (3)$$

For each word e, we then cluster its most similar instances using group-average clustering (Han and Kamber 2001) and we store as a candidate committee the highest scoring cluster c' according to the following metric:

$$|\, c'| \times \text{avgsim}(c') \qquad (4)$$

where $|c'|$ is the number of elements in c' and avgsim(c') is the average pairwise similarity between words in c'. The assumption is that the best representative for a concept is a large set of very similar instances. The committee for class c is then the highest scoring candidate committee containing only words from c. For example, below are the committee members discovered for the semantic classes (A), (B), and (C) from Section 1:

1) cardiovascular disease, diabetes, multiple sclerosis, osteoporosis, Parkinson's, rheumatoid arthritis

2) Curtis Joseph, John Vanbiesbrouck, Mike Richter, Tommy Salo

3) blue, pink, red, yellow

3.3 Phase III

By averaging the feature vectors of the committee members of a particular semantic class, we obtain a grammatical template, or signature, for that class. For example, Figure 1 shows an excerpt of the grammatical signature for concept (B) in Section 1. The vector is obtained by averaging the feature vectors for the words *Curtis Joseph*, *John Vanbiesbrouck*, *Mike Richter*, and *Tommy Salo* (the committee of this concept). The

{Curtis Joseph, John Vanbiesbrouck, Mike Richter, Tommy Salo}		
-N:gen:N		
pad	57	11.19
backup	29	9.95
crease	7	9.69
glove	52	9.57
stick	20	9.15
shutout	17	8.80
-N:conj:N		
Hasek	15	12.36
Martin Brodeur	12	12.26
Belfour	13	12.22
Patrick Roy	10	11.90
Dominik Hasek	7	11.20
Roy	6	10.01
-V:subj:N		
sprawl	11	6.69
misplay	6	6.55
smother	10	6.54
skate	28	6.43
turn back	10	6.28
stop	453	6.19
N:appo:N		
goaltender	449	10.79
goalie	1641	10.76
netminder	57	10.39
goalkeeper	487	9.69
N:conj:N		
Martin Brodeur	11	12.49
Dominik Hasek	11	12.33
Ed Belfour	10	12.04
Curtis Joseph	7	11.46
Tom Barrasso	5	10.85
Byron Dafoe	5	10.80
Chris Osgood	4	10.25

Figure 1. Excerpt of the grammatical signature for the *goalie*/*goaltender* concept.

"*-V:subj:N:sprawl*" feature indicates a subject-verb relationship between the concept and the verb *sprawl* while "*N:appo:N:goaltender*" indicates an apposition relationship between the concept and the noun *goaltender*. The (-) in a relationship means that the right hand side of the relationship is the head (e.g. *sprawl* is the head of the subject-verb relationship). The two columns of numbers indicate the frequency and mutual information score for each feature respectively.

In order to discover the characteristics of human naming conventions, we manually named 50 concepts discovered by CBC. For each concept, we extracted the relationships between the concept committee and the assigned label. We then added the mutual information scores for each extracted relationship among the 50 concepts. The top-4 highest scoring relationships are:

- *Apposition (N:appo:N)*
  e.g. ... **Oracle**, a **company** known for its progressive employment policies, ...
- *Nominal subject (-N:subj:N)*
  e.g. ... **Apple** was a hot young **company**, with Steve Jobs in charge.
- *Such as (-N:such as:N)*
  e.g. ... **companies** such as **IBM** must be weary ...
- *Like (-N:like:N)*
  e.g. ... **companies** like **Sun Microsystems** do no shy away from such challenges, ...

To name a class, we simply search for these syntactic relationships in the signature of a concept. We sum up the mutual information scores for each term that occurs in these relationships with a committee of a class. The highest scoring term is the name of the class. For example, the top-5 scoring terms that occurred in these relationships with the signature of the concept represented by the committee *{Curtis Joseph, John Vanbiesbrouck, Mike Richter, Tommy Salo}* are:

1)	goalie	40.37
2)	goaltender	33.64
3)	goalkeeper	19.22
4)	player	14.55
5)	backup	9.40

The numbers are the total mutual information scores of each name in the four syntactic relationships.

4 Evaluation

In this section, we present an evaluation of the class labeling algorithm and of the hyponym relationships discovered by our system.

4.1 Experimental Setup

We used Minipar (Lin 1994), a broad coverage parser, to parse 3GB of newspaper text from the Aquaint (TREC-9) collection. We collected the frequency counts of the grammatical relationships (contexts) output by Minipar and used them to compute the pointwise mutual information vectors described in Section 3.1.

We used the 1432 noun clusters extracted by CBC[1] as the list of concepts to name. For each concept, we then used our algorithm described in Section 3 to extract the top-20 names for each concept.

[1] Available at http://www.isi.edu/~pantel/demos.htm

Table 1. Labels assigned to 10 randomly selected concepts (each represented by three committee members.

CBC Concept	*Human Label*	*WordNet Labels*	*System Labels* (ranked)
BMG, EMI, Sony	record label	none	label / company / album / machine / studio
Preakness Stakes, Preakness, Belmont Stakes	horse race	none	race / event / run / victory / start
Olympia Snowe, Susan Collins, James Jeffords	US senator	none	republican / senator / chairman / supporter / conservative
Eldoret, Kisumu, Mombasa	African city	none	city / port / cut off / town / southeast
Bronze Star, Silver Star, Purple Heart	medal	decoration / laurel wreath / medal / medallion / palm	distinction / set / honor / symbol
Mike Richter, Tommy Salo, John Vanbiesbrouck	NHL goalie	none	goalie / goaltender / goalkeeper / player / backup
Dodoma, Mwanza, Mbeya	African city	none	facilitator / town
fresco, wall painting, Mural	art	painting / picture	painting / world / piece / floor / symbol
Qinghua University, Fudan University, Beijing University	university	none	university / institution / stockholder / college / school
Federal Bureau of Investigation, Drug Enforcement Administration, FBI	governmental department	law enforcement agency	agency / police / investigation / department / FBI

4.2 Labeling Precision

Out of the 1432 noun concepts, we were unable to name 21 (1.5%) of them. This occurs when a concept's committee members do not occur in any of the four syntactic relationships described in Section 0. We performed a manual evaluation of the remaining 1411 concepts.

We randomly selected 125 concepts and their top-5 highest ranking names according to our algorithm. Table 1 shows the first 10 randomly selected concepts (each concept is represented by three of its committee members).

For each concept, we added to the list of names a human generated name (obtained from an annotator looking at only the concept instances). We also appended concept names extracted from WordNet. For each concept that contains at least five instances in the WordNet hierarchy, we named the concept with the most frequent common ancestor of each pair of instances. Up to five names were generated by WordNet for each concept. Because of the low coverage of proper nouns in WordNet, only 33 of the 125 concepts we evaluated had WordNet generated labels.

We presented to three human judges the 125 randomly selected concepts together with the system, human, and WordNet generated names randomly ordered. That way, there was no way for a judge to know the source of a label nor the system's ranking of the labels. For each name, we asked the judges to assign a score of *correct*, *partially correct*, or *incorrect*. We then computed the mean reciprocal rank (MRR) of the system, human, and WordNet labels. For each concept, a naming scheme receives a score of 1 / M where M is the rank of the first name judged correct. Table 2 shows the results. Table 3 shows similar results for a more lenient evaluation where M is the rank of the first name judged correct or partially correct.

Our system achieved an overall MRR score of 77.1%. We performed much better than the baseline WordNet (19.9%) because of the lack of coverage (mostly proper nouns) in the hierarchy. For the 33 concepts that WordNet named, it achieved a score of 75.3% and a lenient score of 82.7%, which is high considering the simple algorithm we used to extract labels using WordNet.

The Kappa statistic (Siegel and Castellan Jr. 1988) measures the agreements between a set of judges' assessments correcting for chance agreements:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \qquad (5)$$

where $P(A)$ is the probability of agreement between the judges and $P(E)$ is the probability that the judges agree

Table 2. MRR scores for the human evaluation of naming 125 random concepts.

Judge	*Human Labels*	*WordNet Labels*	*System Labels*
1	100%	18.1%	74.4%
2	91.2%	20.0%	78.1%
3	89.6%	21.6%	78.8%
Combined	93.6%	19.9%	77.1%

Table 3. Lenient MRR scores for the human evaluation of naming 125 random concepts.

Judge	*Human Labels*	*WordNet Labels*	*System Labels*
1	100%	22.8%	85.0%
2	96.0%	20.8%	86.5%
3	92.0%	21.8%	85.2%
Combined	96.0%	21.8%	85.6%

Table 4. Percentage of concepts with a correct name in the top-5 ranks returned by our system.

Judge	*Top-1*	*Top-2*	*Top-3*	*Top-4*	*Top-5*
1	68.8%	75.2%	78.4%	83.2%	84.0%
2	73.6%	80.0%	81.6%	83.2%	84.8%
3	73.6%	80.0%	82.4%	84.0%	88.8%
Combined	72.0%	78.4%	80.8%	83.5%	85.6%

Table 5. Accuracy of 159,000 extracted hyponyms and a subset of 65,000 proper noun hyponyms.

Judge	*All Nouns*		*Proper Nouns*	
	Strict	Lenient	Strict	Lenient
1	62.0%	68.0%	79.0%	82.0%
2	74.0%	76.5%	84.0%	85.5%
Combined	68.0%	72.2%	81.5%	83.8%

by chance on an assessment. An experiment with $K \geq 0.8$ is generally viewed as reliable and $0.67 < K < 0.8$ allows tentative conclusions. The Kappa statistic for our experiment is $K = 0.72$.

The human labeling is at a disadvantage since only one label was generated per concept. Therefore, the human scores either 1 or 0 for each concept. Our system's highest ranking name was correct 72% of the time. Table 4 shows the percentage of semantic classes with a correct label in the top 1-5 ranks returned by our system.

Overall, 41.8% of the top-5 names extracted by our system were judged correct. The overall accuracy for the top-4, top-3, top-2, and top-1 names are 44.4%, 48.8%, 58.5%, and 72% respectively. Hence, the name ranking of our algorithm is effective.

4.3 Hyponym Precision

The 1432 CBC concepts contain 18,000 unique words. For each concept to which a word belongs, we extracted up to 3 hyponyms, one for each of the top-3 labels for the concept. The result was 159,000 hyponym relationships. 24 are shown in the Appendix.

Two judges annotated two random samples of 100 relationships: one from all 159,000 hyponyms and one from the subset of 65,000 proper nouns. For each instance, the judges were asked to decide whether the hyponym relationship was *correct*, *partially correct* or *incorrect*. Table 5 shows the results. The strict measure counts a score of 1 for each correctly judged instance and 0 otherwise. The lenient measure also gives a score of 0.5 for each instance judged partially correct.

Many of the CBC concepts contain noise. For example, the *wine* cluster:

```
Zinfandel, merlot, Pinot noir, Chardonnay,
Cabernet  Sauvignon,  cabernet,  riesling,
Sauvignon blanc, Chenin blanc, sangiovese,
syrah, Grape, Chianti ...
```

contains some incorrect instances such as *grape*, *appelation*, and *milk chocolate*. Each of these instances will generate incorrect hyponyms such as *grape is wine* and *milk chocolate is wine*. This hyponym extraction task would likely serve well for evaluating the accuracy of lists of semantic classes.

Table 5 shows that the hyponyms involving proper nouns are much more reliable than common nouns. Since WordNet contains poor coverage of proper nouns, these relationships could be useful to enrich it.

4.4 Recall

Semantic extraction tasks are notoriously difficult to evaluate for recall. To approximate recall, we conducted two question answering (QA) tasks: answering definition questions and performing QA information retrieval.

Table 6. Percentage of correct answers in the Top-1 and Top-5 returned answers on 50 definition questions.

System	*Top-1*		*Top-5*	
	Strict	Lenient	Strict	Lenient
WordNet	38%	38%	38%	38%
Fleischman	36%	40%	42%	44%
Our System	36%	44%	60%	62%

Table 7. Percentage of questions where the passage retrieval module returns a correct answer in the Top-1 and Top-100 ranked passages (with and without semantic indexing).

	Correct Top-1	*Correct Top-100*
With semantic indexing	43 / 179	134 / 179
Without semantic indexing	36 / 179	131 / 179

Definition Questions

We chose the 50 definition questions that appeared in the QA track of TREC2003 (Voorhees, 2003). For example: "*Who is Aaron Copland*?" and "*What is the Kama Sutra*?" For each question we looked for at most five corresponding concepts in our hyponym list. For example, for *Aaron Copland*, we found the following hypernyms: *composer*, *music*, and *gift*. We compared our system with the concepts in WordNet and Fleischman et al.'s instance/concept relations (Fleischman et al. 2003). Table 6 shows the percentage of correct answers in the top-1 and top-5 returned answers from each system. All systems seem to have similar performance on the top-1 answers, but our system has many more answers in the top-5. This shows that our system has comparatively higher recall for this task.

Information (Passage) Retrieval

Passage retrieval is used in QA to supply relevant information to an answer pinpointing module. The higher the performance of the passage retrieval module, the higher will be the performance of the answer pinpointing module.

The passage retrieval module can make use of the hyponym relationships that are discovered by our system. Given a question such as "*What color …*", the likelihood of a correct answer being present in a retrieved passage is greatly increased if we know the set of all possible colors and index them in the document collection appropriately.

We used the hyponym relations learned by our system to perform semantic indexing on a QA passage retrieval task. We selected the 179 questions from the QA track of TREC-2003 that had an explicit semantic answer type (e.g. "*What band was Jerry Garcia with?*" and "*What color is the top stripe on the U.S. flag?*"). For each expected semantic answer type corresponding to a given question (e.g. *band* and *color*), we indexed the entire TREC-2002 IR collection with our system's hyponyms.

We compared the passages returned by the passage retrieval module with and without the semantic indexing. We counted how many of the 179 questions had a correct answer returned in the top-1 and top-100 passages. Table 7 shows the results.

Our system shows small gains in the performance of the IR output. In the top-1 category, the performance improved by 20%. This may lead to better answer selections.

5 Conclusions and Future Work

Current state of the art concept discovery algorithms generate lists of instances of semantic classes but stop short of labeling the classes with concept names. Class labels would serve useful in applications such as question answering to map a question concept into a semantic class and then search for answers within that class. We propose here an algorithm for automatically labeling concepts that searches for syntactic patterns within a grammatical template for a class. Of the 1432 noun concepts discovered by CBC, our system labelled 98.5% of them with an MRR score of 77.1% in a human evaluation.

Hyponym relationships were then easily extracted, one for each instance/concept label pair. We extracted 159,000 hyponyms and achieved a precision of 68%. On a subset of 65,000 proper names, our performance was 81.5%.

This work forms an important attempt to building large-scale semantic knowledge bases. Without being able to automatically name a cluster and extract hyponym/hypernym relationships, the utility of automatically generated clusters or manually compiled lists of terms is limited. Of course, it is a serious open question how many names each cluster (concept) should have, and how good each name is. Our method begins to address this thorny issue by quantifying the name assigned to a class and by simultaneously assigning a number that can be interpreted to reflect the strength of membership of each element to the class. This is potentially a significant step away from traditional all-or-nothing semantic/ontology representations to a concept representation

Appendix. Sample hyponyms discovered by our system.

Instance	*Concept*	*Instance*	*Concept*
actor	hero	price support	benefit
Ameritrade	brokerage	republican	politician
Arthur Rhodes	pitcher	Royal Air Force	force
bebop	MUSIC	Rwanda	city
Buccaneer	team	Santa Ana	city
Congressional Research Service	agency	shot-blocker	player
Cuba	country	slavery	issue
Dan Petrescu	midfielder	spa	facility
Hercules	aircraft	taxi	vehicle
Moscow	city	Terrence Malick	director
Nokia	COMPANY	verbena	tree
nominee	candidate	Wagner	composer

scheme that is more nuanced and admits multiple names and graded set memberships.

Acknowledgements

The authors wish to thank the reviewers for their helpful comments. This research was partly supported by NSF grant #EIA-0205111.

References

Barwise, J. and Perry, J. 1985. Semantic innocence and uncompromising situations. In: Martinich, A. P. (ed.) The Philosophy of Language. New York: Oxford University Press. pp. 401–413.

Berland, M. and E. Charniak, 1999. Finding parts in very large corpora. In *ACL-1999*. pp. 57–64. College Park, MD.

Church, K. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of ACL-89*. pp. 76–83. Vancouver, Canada.

Fleischman, M.; Hovy, E.; and Echihabi, A. 2003. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of ACL-03*. pp. 1–7. Sapporo, Japan.

Girju, R.; Badulescu, A.; and Moldovan, D. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of HLT/NAACL-03*. pp. 80–87. Edmonton, Canada.

Han, J. and Kamber, M. 2001. *Data Mining – Concepts and Techniques*. Morgan Kaufmann.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In COLING-92. pp. 539–545. Nantes, France.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*. pp. 268–275. Pittsburgh, PA.

Leacock, C.; Chodorow, M.; and Miller; G. A. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

Lenat, D. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Lin, D. 1994. Principar - an efficient, broad-coverage, principle-based parser. *Proceedings of COLING-94*. pp. 42–48. Kyoto, Japan.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98*. pp. 768–774. Montreal, Canada.

Lin, D. and Pantel, P. 2001. Induction of semantic classes from natural language text. In *Proceedings of SIGKDD-01*. pp. 317–322. San Francisco, CA.

Mann, G. S. 2002. Fine-Grained Proper Noun Ontologies for Question Answering. *SemaNet' 02: Building and Using Semantic Networks*, Taipei, Taiwan.

Miller, G. 1990. WordNet: An online lexical database. *International Journal of Lexicography*, 3(4).

Pasca, M. and Harabagiu, S. 2001. The informative role of WordNet in Open-Domain Question Answering. In *Proceedings of NAACL-01 Workshop on WordNet and Other Lexical Resources*. pp. 138–143. Pittsburgh, PA.

Pantel, P. and Lin, D. 2002. Discovering Word Senses from Text. In *Proceedings of SIGKDD-02*. pp. 613–619. Edmonton, Canada.

Riloff, E. and Shepherd, J. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of EMNLP-1997*.

Riloff, E. and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of AAAI-99*. pp. 474–479. Orlando, FL.

Salton, G. and McGill, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill

Schank, R. and Abelson, R. 1977. Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures. Lawrence Erlbaum Associates.

Siegel, S. and Castellan Jr., N. J. 1988. Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill.

Voorhees, E. 2003. Overview of the question answering track. To appear in *Proceedings of TREC-12 Conference*. NIST, Gaithersburg, MD.