Learning syntactic patterns for automatic hypernym discovery

DRAFT VERSION - DO NOT CIRCULATE

Rion Snow
Computer Science Department
Stanford University
Stanford, CA 94305
rion@cs.stanford.edu

Daniel Jurafsky
Linguistics Department
Stanford University
Stanford, CA 94305
jurafsky@stanford.edu

Andrew Y. Ng
Computer Science Department
Stanford University
Stanford, CA 94305
ang@cs.stanford.edu

Abstract

We present a new algorithm for learning hypernym (is-a) relations from text, a key problem in machine learning for natural language understanding. This method generalizes earlier work that relied on hand-built lexico-syntactic patterns by introducing a general-purpose formalization of the pattern space based on syntactic dependency paths. We learn these paths automatically by taking hypernym/hyponym word pairs from WordNet, finding sentences containing these words in a large parsed corpus, and automatically extracting these paths. These paths are then used as features in a high-dimensional representation of noun relationships. We use a logistic regression classifier based on these features for the task of corpus-based hypernym pair identification. Our classifier is shown to outperform previous pattern-based methods for identifying hypernym pairs (using WordNet as a gold standard), and is shown to outperform those methods as well as WordNet on an independent test set.

1 Introduction

Semantic taxonomies and thesauri like WordNet [5, 13] are a key source of knowledge for natural language processing applications, giving structured information about semantic relations between words. Building such taxonomies, however, is an extremely slow and knowledge-intensive process, and furthermore any particular semantic taxonomy is bound to be limited in its scope and domain. Thus a wide variety of recent research has focused on finding methods for automatically learning taxonomic relations and constructing semantic hierarchies [1, 2, 3, 4, 6, 8, 9, 10, 16, 18, 19, 20, 21, 22].

In this paper we focus on building an automatic classifier for the HYPERNYM/HYPONYM relation. A word X is a hyponym of word Y if X is a subtype or instance of Y. Thus 'Shakespeare' is a HYPONYM of 'author', (and conversely 'author' is a HYPERNYM of 'Shakespeare') 'dog' is a hyponym of 'canine', 'table' is a hyponym of 'furniture', and so on.

Much of the previous research on automatic semantic classification of words has focused on a key insight first articulated by Hearst in [9], that the presence of certain 'lexico-syntactic patterns' can indicate a particular semantic relationship between two nouns. Hearst noticed, for example, that linking two noun phrases (NPs) via the constructions "Such NP_Y as NP_X", or "NP_X and other NP_Y", often implies the relation *hyponym*(NP_X, NP_Y), i.e. that NP_X is a kind of NP_Y. Since then, a broad swath of researchers has used a small number (typically less than 10) of hand-created patterns like those of Hearst to automatically label such semantic relations [1, 2, 6, 18, 19]. While these patterns have been

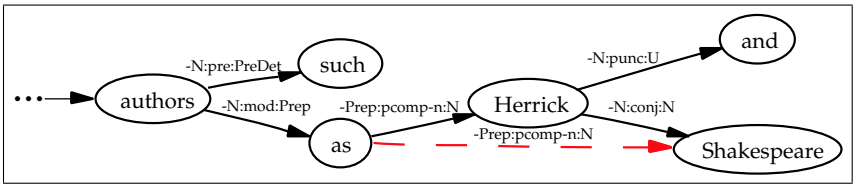

Figure 1: MINIPAR dependency tree example with transform

successful at identifying some examples of relationships like hypernymy, this method of lexicon construction is tedious and subject to the bias of the designer; further, such pattern lexicons contain only a small subset of the actual 'patterns' found to occur in natural text.

Our goal is to use a machine learning paradigm to automatically replace this hand-built knowledge. In our new approach to the hypernym-labeling task, based on extending a suggestion from [9], patterns indicative of hypernymy are learned automatically under "indirect" or "distant supervision" from a thesaurus, as follows:

1. Training:
   (a) Extract examples of all hypernym pairs (pairs of words in a hypernym/hyponym relation) from WordNet.
   (b) For each hypernym pair, find sentences in which both words occur.
   (c) Parse the sentences, and automatically extract patterns from the parse tree which are good cues for hypernymy.
   (d) Train a hypernym classifier based on these features.
2. Test:
   (a) Given a pair of words in the test set, extract features and use the classifier to decide if the word-pair is in the hypernym/hyponym relation or not.

The next section introduces our method for automatically discovering patterns indicative of hypernymy. Section 3 then describes the setup of our experiments. In Section 4 we analyze our feature space, and in Section 5 we describe a combined classifier based on these features which achieves high accuracy at the task of hypernym identification. Section 6 shows how this classifier can be improved by adding a new source of knowledge, coordinate terms.

2 Representing lexico-syntactic patterns with *dependency paths*

The first goal of our work is to automatically identify lexico-syntactic patterns indicative of hypernymy. In order to do this, we need a representation space for expressing these patterns. We propose the use of *dependency paths* as a general-purpose formalization of the space of lexico-syntactic patterns, based on the broad-coverage dependency parser MINIPAR [11]. Dependency paths have been used successfully in the past to represent lexico-syntactic relations suitable for semantic processing [12].

A dependency parser produces a dependency tree that represents the syntactic relations between words by a list of edge tuples of the form:
($word_1$,CATEGORY$_1$:RELATION:CATEGORY$_2$, $word_2$). Here each $word$ is the stemmed form of the word or multi-word phrase (so that "*authors*" becomes "*author*"), and corresponds to a specific node in the dependency tree; each $category$ is the part of speech label of the corresponding word (e.g. N for noun or PREP for preposition); and the $relation$ is the directed syntactic relationship exhibited from $word_1$ to $word_2$ (e.g. OBJ for object, MOD for modifier, or CONJ for conjunct), and corresponds to a specific link in the tree. We may then define our space of lexico-syntactic patterns to be all shortest paths of four links or less between any two nouns in a dependency tree. Figure 2 shows the partial dependency tree for the sentence fragment *"...such authors as Herrick and Shakespeare"*.

We then remove the original words in the noun pair to create a more general pattern. Each dependency path may then be presented as an ordered list of dependency tuples. We extend

NP_X and other NP_Y:	(*and*,U:PUNC:N),-N:CONJ:N, (*other*,A:MOD:N)
NP_X or other NP_Y:	(*or*,U:PUNC:N),-N:CONJ:N, (*other*,A:MOD:N)
NP_Y such as NP_X:	N:PCOMP-N:PREP,*such_as*,*such_as*,PREP:MOD:N
Such NP_Y as NP_X:	N:PCOMP-N:PREP,*as*,*as*,PREP:MOD:N,(*such*,PREDET:PRE:N)
NP_Y including NP_X:	N:OBJ:V,*include*,*include*,V:I:C,*dummy_node*,*dummy_node*,C:REL:N
NP_Y, especially NP_X:	-N:APPO:N,(*especially*,A:APPO-MOD:N)

Table 1: Dependency path representations of Hearst's patterns

this basic MINIPAR representation in two ways: first, we wish to capture the fact that certain function words like 'such' (in 'such NP as NP') or 'other' (in 'NP and other NPs') are important parts of lexico-syntactic patterns. We implement this by adding optional "satellite links" to each shortest path, i.e. single links not already contained in the dependency path added on either side of each noun. Second, we capitalize on the distributive nature of the syntactic "conjunct" relation (e.g. "and","or", and comma-separated noun lists) by distributing dependency links across such conjuncts. As an example, in the simple 2-member conjunct chain of *Herrick* and *Shakespeare* in Figure 2, we add the entrance link "*as*, -PREP:PCOMP-N:N" to the single element '*Shakespeare*' (as a dotted line in the figure). Our extended dependency notation is able to capture the power of the hand-engineered patterns described in the literature. Table 1 shows the six patterns used in [1, 2, 9] and their corresponding dependency path formalizations.

3 Experimental paradigm

Our goal is to build a classifier which is given an ordered pair of words and makes a binary decision as to whether the nouns are related by hypernymy or not.

All of our experiments are based on a corpus of over 6 million newswire sentences.[1] We first parsed each of the sentences in the corpus using MINIPAR. We extract every pair of nouns from each sentence.

752,311 of the resulting unique noun pairs were labeled as Known Hypernym or Known Non-Hypernym using WordNet[2]. A noun pair (n_1, n_2) is labeled Known Hypernym if n_2 is an ancestor of the first sense of n_1 in the WordNet hypernym taxonomy, and if the only "frequently-used" [3] sense of each word is the first noun sense listed in WordNet. Note that n_2 is considered a hypernym of n_1 regardless of how much higher in the hierarchy it is with respect to n_1. A noun pair may be assigned to the second set of Known Non-Hypernym pairs if both nouns are contained within WordNet, but neither word is an ancestor of the other in the WordNet hypernym taxonomy for any senses of either word. Of our collected noun pairs, 14,387 were Known Hypernym pairs, and we assign the 737,924 most frequently occurring Known Non-Hypernym pairs to the second set; this number is selected to preserve the roughly 1:50 ratio of hypernym-to-non-hypernym pairs observed in our hand-labeled test set (discussed below).

We evaluated our binary classifiers in two ways. For both sets of evaluations, our classifier was given a pair of words from an unseen sentence and had to make a hypernym vs. non-hypernym decision. In the first style of evaluation, we compared the performance of our classifiers against the Known Hypernym versus Known Non-Hypernym labels assigned by WordNet. This provides a metric for how well our classifiers do at "recreating" WordNet.

For the second set of evaluations we hand-labeled a test set of 5,387 noun pairs from randomly-selected paragraphs within our corpus (with part-of-speech labels assigned by MINIPAR). The annotators are instructed to label each ordered noun pair as one of

[1] The corpus contains articles from the Associated Press, Wall Street Journal, and Los Angeles Times, drawn from the TIPSTER 1, 2, 3, and TREC 5 corpora [7].

[2] We access WordNet 2.0 via Jason Rennie's WordNet::QueryData interface.

[3] A noun sense is determined to be "frequently-used" if it occurs at least once in the sense-tagged Brown Corpus Semantic Concordance files (as reported in the `cntlist` file distributed as part of WordNet 2.0). This determination is made so as to reduce the number of false hypernym/hyponym classifications due to highly polysemous words.

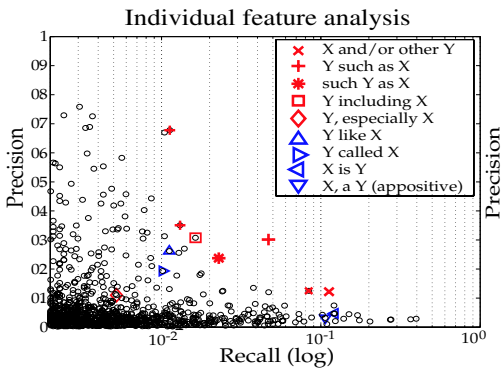

Figure 2: Hypernym pre/re for all features

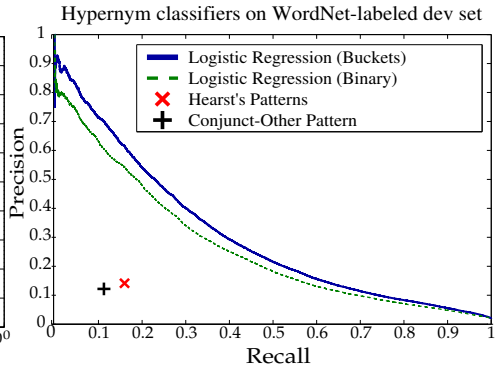

Figure 3: Hypernym classifiers

"hyponym-to-hypernym", "hypernym-to-hyponym", "coordinate", or "unrelated" (the coordinate relation will be defined below). As expected, the vast majority of pairs (5,122) were found to be unrelated by these measures; the rest were split evenly between hypernym and coordinate pairs (134 and 131, resp.).

Interannotator agreement was obtained between four labelers (all native speakers of English) on a held-out set of 511 noun pairs, and determined for each task according to the averaged F-Score across all pairs of the four labelers. Agreement was 83% and 64% for the hypernym and coordinate term classification tasks, respectively.

4 Features: pattern discovery

Our first study focused on discovering which dependency paths (lexico-syntactic patterns) might prove useful features for our classifiers. To evaluate these features, we construct a binary classifier for each pattern, which simply classifies a noun pair as hypernym/hyponym if and only if the specific pattern occurs at least once for that noun pair. Figure 2 depicts the precision and recall of all such classifiers (with recall at least .0015) on the WordNet-labeled data set[4]. Using this formalism we have been able to capture a wide variety of repeatable patterns between hypernym/hyponym noun pairs; in particular, we have been able to 'rediscover' the hand-designed patterns originally proposed in [9] (the first five features, marked in red[5]), in addition to a number of new patterns not previously discussed (of which four are marked as blue triangles in Figure 2 and listed in Table 2. This analysis gives a quantitative justification to Hearst's initial intuition as to the power of hand-selected patterns; nearly all of Hearst's patterns are at the high-performance boundary of precision and recall for individual features.

NP_Y like NP_X:	N:PCOMP-N:PREP,*like*,*like*,PREP:MOD:N
NP_Y called NP_X:	N:DESC:V,*call*,*call*,V:VREL:N
NP_X is a NP_Y:	N:S:VBE,*be*,*be*,-VBE:PRED:N
NP_X, a NP_Y (appositive):	N:APPO:N

Table 2: Dependency path representations of other high-scoring patterns

5 A hypernym-only classifier

Our first hypernym classifier is based on the intuition that unseen noun pairs are likely to be in a hypernymy relation if they occur in the test set in one or more lexico-syntactic patterns indicative of hypernymy.

[4]Redundant features consisting of an identical base path to an identified pattern but differing only by an additional "satellite link" are marked in Figure 2 by smaller versions of the same symbol.

[5]We mark the single generalized "*conjunct* other" pattern -N:CONJ:N, (*other*,A:MOD:N) to represent both of Hearst's original "and other" and "or other" patterns

Best Logistic Regression (Buckets):	0.3480
Best Logistic Regression (Binary):	0.3200
Best Multinomial Naive Bayes:	0.3175
Best Complement Naive Bayes:	0.3024
Hearst Patterns:	0.1500
Caraballo Pattern:	0.1170

Table 3: Average maximum F-score for cross validation on WordNet-labeled training set

From the 6 million word corpus, we created a 'feature lexicon' which contained each dependency path that occurred between at least five unique noun pairs in our corpus. This results in a feature lexicon of approximately 70,000 dependency paths. Next, we record in our noun pair lexicon each noun pair that occurs within our corpus with at least five unique paths from this lexicon. We then create a feature count vector for each noun pair. Each dimension of the 69,592-dimension vector represents a particular dependency path, and contains the total number of times in our corpus that that path was the shortest path connecting that noun pair in some dependency tree.

We thus define as our task the binary classification of noun pair hypernymy or non-hypernymy based on its feature vector of dependency paths.

We use the WordNet-labeled Known Hypernym / Known Not-Hypernym training set defined in the previous section. We train a variety of classifiers on this data set, including multinomial Naive Bayes, complement Naive Bayes [17], and logistic regression. We perform model selection using 10-fold cross validation on this training set, evaluating each model based on its maximum hypernym F-Score averaged across all folds. The summary of average maximum F-scores is presented in Table 3, and the precision/recall plot of our best models is presented in Figure 3. For comparison, we evaluate two simple classifiers based on past work with a handful of hand-engineered features; the first simply detects the presence of at least one of Hearst's pattern, arguably the previous best classifier consisting only of lexico-syntactic patterns, and as implemented for hypernym discovery in [2]. The second classifier consists of only the "NP and/or other NP" subset of Hearst's patterns, as used in the automatic construction of noun-labeled hypernym taxonomies in [1]. In our tests we found greatest performance from a binary logistic regression model with 14 redundant threshold buckets spaced at the exponentially increasing intervals $\{1, 2, 4, ...4096, 8192\}$; our resulting feature space consists of 923,328 distinct binary features. These buckets are defined such that a feature corresponding to pattern p at threshold t will be activated by a noun pair n if and only if p has been observed to occur as a shortest dependency path between n at least t times.

Our classifier shows a dramatic improvement over previous classifiers; in particular, using our best logistic regression classifier, we observe a 132% relative improvement of average maximum F-score over the classifier based on Hearst's patterns.

6 Using Coordinate Terms to Improve Hypernym Classification

While our hypernym-only classifier performed better than previous classifiers based on hand-built patterns, there is still much room for improvement. As [2] point out, one problem with pattern-based hypernym classifiers in general is that within-sentence hypernym pattern information is quite sparse. Patterns are useful only to classify noun pairs which happen to occur in the same sentence; many hypernym/hyponym pairs may simply not occur in the same sentence in the corpus. For this reason [2], following [1] suggest relying on a second source of knowledge: 'coordinate' relations between words. The *coordinate term* relation is defined in the WordNet glossary as: "Y is a coordinate term of X if X and Y share a hypernym." The coordinate relation is a symmetric relation between words that are "the same kind of thing", i.e. that share at least one common ancestor in the hypernym taxonomy. Many methods exist for inferring that two words are coordinate term (a common subtask in automatic thesaurus induction). Thus we expect that using coordinate information might increase the recall of our hypernym classifier: if we are confident

Interannotator Average:	0.6405
Distributional Similarity Vector Space Model for :	0.3327
Thresholded Conjunct Classifier:	0.2857
Best WordNet F-score:	0.2630

Table 4: Summary of maximum F-scores on hand-labeled coordinate pairs

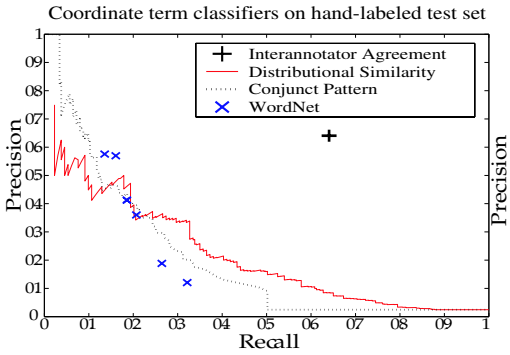

Figure 4: Coordinate classifiers on hand-labeled test set

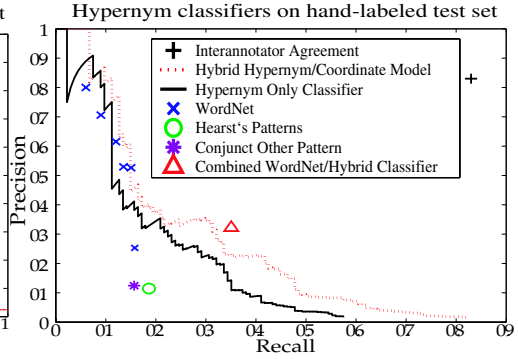

Figure 5: Hypernym classifiers on hand-labeled test set

that two entities e_i, e_j are coordinate terms, and that e_j is a hyponym of e_k, we may then infer with higher probability that e_i is similarly a hyponym of e_k – despite never having encountered the pair (e_i, e_k) within a single sentence.

6.1 Coordinate Term Classification

Prior work for classifying the coordinate relation include automatic word sense clustering methods based on *distributional* similarity (e.g. [14, 15]) or on pattern-based techniques, specifically using the coordination pattern 'X, Y, and Z' (e.g. [2]). We construct both types of classifier. First we construct a vector-space model similar to [14] using single MINIPAR dependency links as our distributional features. Using the same 6 million MINIPAR-parsed sentences used in our hypernym training set, we first construct a feature lexicon of the 30,000 most frequent single dependency edges summed across all edges connected to any noun in our corpus; we then construct feature count vectors for each of the most frequently occurring 163,198 individual nouns. We normalize these feature counts with pointwise mutual information, and compute as our measure of similarity the cosine coefficient between these normalized vectors. We evaluate this classifier on our hand-labeled test set, where of 5,387 total pairs, 131 are labeled as "coordinate". For purposes of comparison we construct a series of classifiers from WordNet, which makes the simple binary decision of determining whether two words are coordinate according to whether they share a common ancestor within n words higher up in the hypernym taxonomy, for all n from 1 to 6. Also, we compare a simple pattern-based classifier based on the *conjunct* pattern (e.g. "X and Y"), which thresholds simply on the number of conjunct patterns found between a given pair. Results of this experiment are shown in Table 4 and Figure 4.

The strong performance of the simple conjunct pattern model suggests that it may be worth pursuing an extended pattern-based coordinate classifier along the lines of our hypernym classifier; for now, we proceed with our simple distributional similarity vector space model (with a 16% relative F-score improvement over the conjunct model) in the construction of a combined hypernym-coordinate hybrid classifier.

6.2 Hybrid hypernym-coordinate classification

Finally we would like to combine our hypernym and coordinate models in order to improve hypernym classification. Thus we define two probabilities of pair relationships between

Interannotator Agreement:	0.8318
Combined WordNet/Hypernym/Coordinate Model:	0.3357
Combined Linear Interpolation Hypernym/Coordinate Model:	0.3268
Best Hypernym-only Classifier (Logistic Regression):	0.2714
Best WordNet F-Score:	0.2339
Hearst Pattern Classifier:	0.1417
And/Or Other Pattern Classifier:	0.1386

Table 5: Final evaluation of hypernym classification on hand-labeled test set

entities: $P(e_i \underset{H}{<} e_j)$ and $P(e_i \underset{C}{\sim} e_j)$, representing the probabilities that entity e_i has e_j as an ancestor in its hypernym hierarchy, and that entities e_i and e_j are *coordinate terms*, i.e. that they share a common hypernym ancestor at some level, respectively. Defining the probability produced by our best hypernym-only classifier as $P_{old}(e_i \underset{H}{<} e_k)$, and a probability score obtained by normalizing the similarity score from our coordinate classifier as $P(e_i \underset{C}{\sim} e_j)$, we apply a simple linear interpolation scheme to compute a new hypernymy probability; specifically, for each pair of entities (e_i, e_k), we recompute the probability that e_k is a hypernym of e_i as:

$$P_{new}(e_i \underset{H}{<} e_k) = \lambda_1 P_{old}(e_i \underset{H}{<} e_k) + \lambda_2 \sum_j P_{old}(e_i \underset{C}{\sim} e_j) P(e_j \underset{H}{<} e_k)$$

We constrain our parameters λ_1, λ_2 such that $\lambda_1 + \lambda_2 = 1$, and then set these parameters using 10-fold cross-validation on our hand-labeled test set. For our final evaluation we use $\lambda_1 = 0.7$.

Our hand-labeled dataset allows us to compare the performance of our classifier directly against WordNet itself. Figure 5 contains a plot of precision / recall vs. WordNet, as well as the methods in the previous comparison, now using the human labels as ground truth.

We compared multiple classifiers based on the WordNet hypernym taxonomy, using a variety of parameters including maximum number of senses of a hyponym to find hypernyms for, maximum distance between the hyponym and its hypernym in the WordNet taxonomy, and whether or not to allow synonyms. The best WordNet-based results are plotted in Figure 5; the model achieving the maximum F-score uses only the first sense of a hyponym, allows a maximum distance of 4 between a hyponym and hypernym, and allows any member of a hypernym synset to be a hypernym. Our logistic regression hypernym-only model has a 16% relative maximum F-score improvement over the best WordNet classifier, while the combined Hypernym/Coordinate model has a 40% relative maximum F-score improvement, and a combined WordNet/Hybrid model (a simple AND of the two classifiers) has a 43% improvement.

In Table 6 we analyze the disagreements between the highest F-score WordNet classifier and our combined hypernym/coordinate classifier. There are 31 such disagreements, with WordNet agreeing with the human labels on 5 and our hybrid model agreeing on the other 26. Here we inspect the types of noun pairs where our model improves upon WordNet, and find that at least 30% of our model's improvements are not restricted to Named Entities; given that the distribution of Named Entities among the labeled hypernyms in our test set is over 60%, this leads us to expect that our classifier will perform well at the task of hypernym induction in more general, non-newswire domains.

7 Conclusions

Our experiments demonstrate that automatic methods can be competitive with WordNet for the identification of hypernym pairs in newswire corpora. In future work we plan to apply our technique to other general knowledge corpora. Further, we plan on extending our algorithms to automatically generate flexible, statistically-grounded hypernym taxonomies directly from corpora.

Type of Noun Pair	Count	Example Pair
Named Entity: Person	7	“John F. Kennedy / president”, “Marlin Fitzwater / spokesman”
Named Entity: Place	7	“Diamond Bar / city”, “France / place”
Named Entity: Company	2	“American Can / company”, “Simmons / company”
Named Entity: Other	1	“Is Elvis Alive / book”
Not Named Entity:	9	“earthquake / disaster”, “soybean / crop”

Table 6: Analysis of improvements over WordNet

Acknowledgments

Thanks to Kayur Patel, Mona Diab, Dan Klein, Allison Buckley, and Todd Huffman for useful discussions and assistance annotating data. Rion Snow is supported by an NDSEG Fellowship sponsored by the DOD and AFOSR.

References

[1] Caraballo, S.A. (2001) Automatic Acquisition of a Hypernym-Labeled Noun Hierarchy from Text. Brown University Ph.D. Thesis.

[2] Cederberg, S. & Widdows, D. (2003) Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. In *Proc. of CoNLL-2003*, pp. 111–118.

[3] Ciaramita, M. & Johnson, M. (2003) Supersense Tagging of Unknown Nouns in WordNet. In *Proc. of EMNLP-2003*.

[4] Ciaramita, M., Hofmann, T., & Johnson, M. (2003) Hierarchical Semantic Classification: Word Sense Disambiguation with World Knowledge. In *Proc. of IJCAI-2003*.

[5] Fellbaum, C. (1998) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

[6] Girju, R., Badulescu A., & Moldovan D. (2003) Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations. In *Proc. of HLT-2003*.

[7] Harman, D. (1992) The DARPA TIPSTER project. *ACM SIGIR Forum* **26**(2), Fall, pp. 26–28.

[8] Hasegawa, T., Sekine, S., & Grishman, R. (2004) Discovering Relations among Named Entities from Large Corpora. In *Proc. of ACL-2004*, pp. 415–422.

[9] Hearst, M. (1992) Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proc. of the Fourteenth International Conference on Computational Linguistics, Nantes, France*.

[10] Hearst, M. & Schütze, H. (1993) Customizing a lexicon to better suit a computational task. In *Proc. of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*.

[11] Lin, D. (1998) Dependency-based Evaluation of MINIPAR. *Workshop on the Evaluation of Parsing Systems, Granada, Spain*

[12] Lin, D. & Pantel P. (2001) Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, **7**(4), pp. 343–360.

[13] Miller, G. (1995) WordNet: a lexical database for English. *Communications of the ACM*

[14] Pantel, P. (2003) Clustering by Committee. Ph.D. Dissertation. Department of Computing Science, University of Alberta.

[15] Pereira, F., Tishby, N., & Lee, L. (1993) Distributional Clustering of English Words. In *Proc. of ACL-1993*, pp. 183–190.

[16] Ravichandran, D. & Hovy, E. (2002) Learning Surface Text Patterns for a Question Answering system. In *Proc. of ACL-2002*.

[17] Rennie J., Shih, L., Teevan, J., & Karger, D. (2003) Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proc. of ICLM-2003*.

[18] Riloff, E. & Shepherd, J. (1997) A Corpus-Based Approach for Building Semantic Lexicons. In *Proc of EMNLP-1997*.

[19] Roark, B. & Charniak, E. (1998) Noun-phrase co-occurerence statistics for semi-automatic-semantic lexicon construction. *Proc. of ACL-1998*, 1110–1116.

[20] Tseng, H. (2003) Semantic classification of unknown words in Chinese. In *Proc. of ACL-2003*.

[21] Turney, P.D., Littman, M.L., Bigham, J. & Shanyder, V. (2003) Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proc. of RANLP-2003*, pp. 482–489.

[22] Widdows, D. (2003) Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proc. of HLT/NAACL 2003*, pp. 276–283.