Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words

Dmitry Davidov
ICNC
The Hebrew University
Jerusalem 91904, Israel
dmitry@alice.nc.huji.ac.il

Ari Rappoport
Institute of Computer Science
The Hebrew University
Jerusalem 91904, Israel
www.cs.huji.ac.il/∼arir

Abstract

We present a novel approach for discovering word categories, sets of words sharing a significant aspect of their meaning. We utilize meta-patterns of high-frequency words and content words in order to discover pattern candidates. Symmetric patterns are then identified using graph-based measures, and word categories are created based on graph clique sets. Our method is the first pattern-based method that requires no corpus annotation or manually provided seed patterns or words. We evaluate our algorithm on very large corpora in two languages, using both human judgments and WordNet-based evaluation. Our fully unsupervised results are superior to previous work that used a POS tagged corpus, and computation time for huge corpora are orders of magnitude faster than previously reported.

1 Introduction

Lexical resources are crucial in most NLP tasks and are extensively used by people. Manual compilation of lexical resources is labor intensive, error prone, and susceptible to arbitrary human decisions. Hence there is a need for automatic authoring that would be as unsupervised and language-independent as possible.

An important type of lexical resource is that given by grouping words into categories. In general, the notion of a *category* is a fundamental one in cognitive psychology (Matlin, 2005). A *lexical category* is a set of words that share a significant aspect of their meaning, e.g., sets of words denoting vehicles, types of food, tool names, etc. A word can obviously belong to more than a single category. We will use ‘category’ instead of ‘lexical category’ for brevity[1].

Grouping of words into categories is useful in itself (e.g., for the construction of thesauri), and can serve as the starting point in many applications, such as ontology construction and enhancement, discovery of verb subcategorization frames, etc.

Our goal in this paper is a fully unsupervised discovery of categories from large unannotated text corpora. We aim for categories containing single words (multi-word lexical items will be dealt with in future papers.) Our approach is based on patterns, and utilizes the following stages:

1. Discovery of a set of pattern candidates that might be useful for induction of lexical relationships. We do this in a fully unsupervised manner, using meta-patterns comprised of *high frequency words* and *content words.*

2. Identification of pattern candidates that give rise to *symmetric* lexical relationships. This is done using simple measures in a word relationship graph.

3. Usage of a novel graph clique-set algorithm in order to generate categories from information on the co-occurrence of content words in the symmetric patterns.

We performed a thorough evaluation on two English corpora (the BNC and a 68GB web corpus) and on a 33GB Russian corpus, and a sanity-check test on smaller Danish, Irish and Portuguese corpora. Evaluations were done using both human

[1] Some people use the term ‘concept’. We adhere to the cognitive psychology terminology, in which ‘concept’ refers to the mental representation of a category (Matlin, 2005).

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 297–304,
Sydney, July 2006. ©2006 Association for Computational Linguistics

judgments and WordNet in a setting quite similar to that done (for the BNC) in previous work. Our unsupervised results are superior to previous work that used a POS tagged corpus, are less language dependent, and are very efficient computationally[2].

Patterns are a common approach in lexical acquisition. Our approach is novel in several aspects: (1) we discover patterns in a fully unsupervised manner, as opposed to using a manually prepared pattern set, pattern seed or words seeds; (2) our pattern discovery requires no annotation of the input corpus, as opposed to requiring POS tagging or partial or full parsing; (3) we discover general symmetric patterns, as opposed to using a few hard-coded ones such as 'x and y'; (4) the clique-set graph algorithm in stage 3 is novel. In addition, we demonstrated the relatively language independent nature of our approach by evaluating on very large corpora in two languages[3].

Section 2 surveys previous work. Section 3 describes pattern discovery, and Section 4 describes the formation of categories. Evaluation is presented in Section 5, and a discussion in Section 6.

2 Previous Work

Much work has been done on lexical acquisition of all sorts. The three main distinguishing axes are (1) the type of corpus annotation and other human input used; (2) the type of lexical relationship targeted; and (3) the basic algorithmic approach. The two main approaches are pattern-based discovery and clustering of context feature vectors.

Many of the papers cited below aim at the construction of hyponym (is-a) hierarchies. Note that they can also be viewed as algorithms for category discovery, because a subtree in such a hierarchy defines a lexical category.

A first major algorithmic approach is to represent word contexts as vectors in some space and use similarity measures and automatic clustering in that space (Curran and Moens, 2002). Pereira (1993) and Lin (1998) use syntactic features in the vector definition. (Pantel and Lin, 2002) improves on the latter by clustering by committee. Caraballo (1999) uses conjunction and appositive annotations in the vector representation.

The only previous works addressing our problem and not requiring any syntactic annotation are those that decompose a lexically-defined matrix (by SVD, PCA etc), e.g. (Schütze, 1998; Deerwester et al, 1990). Such matrix decomposition is computationally heavy and has not been proven to scale well when the number of words assigned to categories grows.

Agglomerative clustering (e.g., (Brown et al, 1992; Li, 1996)) can produce hierarchical word categories from an unannotated corpus. However, we are not aware of work in this direction that has been evaluated with good results on lexical category acquisition. The technique is also quite demanding computationally.

The second main algorithmic approach is to use lexico-syntactic patterns. Patterns have been shown to produce more accurate results than feature vectors, at a lower computational cost on large corpora (Pantel et al, 2004). Hearst (1992) uses a manually prepared set of initial lexical patterns in order to discover hierarchical categories, and utilizes those categories in order to automatically discover additional patterns.

(Berland and Charniak, 1999) use hand crafted patterns to discover part-of (meronymy) relationships, and (Chklovski and Pantel, 2004) discover various interesting relations between verbs. Both use information obtained by parsing. (Pantel et al, 2004) reduce the depth of the linguistic data used but still requires POS tagging.

Many papers directly target specific applications, and build lexical resources as a side effect. Named Entity Recognition can be viewed as an instance of our problem where the desired categories contain words that are names of entities of a particular kind, as done in (Freitag, 2004) using co-clustering. Many Information Extraction papers discover relationships between words using syntactic patterns (Riloff and Jones, 1999).

(Widdows and Dorow, 2002; Dorow et al, 2005) discover categories using two hard-coded symmetric patterns, and are thus the closest to us. They also introduce an elegant graph representation that we adopted. They report good results. However, they require POS tagging of the corpus, use only two hard-coded patterns ('x and y', 'x or y'), deal only with nouns, and require non-trivial computations on the graph.

A third, less common, approach uses set-theoretic inference, for example (Cimiano et al,

[2] We did not compare against methods that use richer syntactic information, both because they are supervised and because they are much more computationally demanding.

[3] We are not aware of any multilingual evaluation previously reported on the task.

2005). Again, that paper uses syntactic information.

In summary, no previous work has combined the accuracy, scalability and performance advantages of patterns with the fully unsupervised, unannotated nature possible with clustering approaches. This severely limits the applicability of previous work on the huge corpora available at present.

3 Discovery of Patterns

Our first step is the discovery of patterns that are useful for lexical category acquisition. We use two main stages: discovery of pattern candidates, and identification of the symmetric patterns among the candidates.

3.1 Pattern Candidates

An examination of the patterns found useful in previous work shows that they contain one or more very frequent word, such as ‘and’, ‘is’, etc. Our approach towards unsupervised pattern induction is to find such words and utilize them.

We define a *high frequency word (HFW)* as a word appearing more than T_H times per million words, and a *content word (CW)* as a word appearing less than T_C times per a million words[4].

Now define a *meta-pattern* as any sequence of HFWs and CWs. In this paper we require that meta-patterns obey the following constraints: (1) at most 4 words; (2) exactly two content words; (3) no two consecutive CWs. The rationale is to see what can be achieved using relatively short patterns and where the discovered categories contain single words only. We will relax these constraints in future papers. Our meta-patterns here are thus of four types: CHC, CHCH, CHHC, and HCHC.

In order to focus on patterns that are more likely to provide high quality categories, we removed patterns that appear in the corpus less than T_P times per million words. Since we can ensure that the number of HFWs is bounded, the total number of pattern candidates is bounded as well. Hence, this stage can be computed in time linear in the size of the corpus (assuming the corpus has been already pre-processed to allow direct access to a word by its index.)

3.2 Symmetric Patterns

Many of the pattern candidates discovered in the previous stage are not usable. In order to find a usable subset, we focus on the symmetric patterns. Our rationale is that two content-bearing words that appear in a symmetric pattern are likely to be semantically similar in some sense. This simple observation turns out to be very powerful, as shown by our results. We will eventually combine data from several patterns and from different corpus windows (Section 4.)

For identifying symmetric patterns, we use a version of the graph representation of (Widdows and Dorow, 2002). We first define the *single-pattern graph* $G(P)$ as follows. Nodes correspond to content words, and there is a directed arc $A(x, y)$ from node x to node y iff (1) the words x and y both appear in an instance of the pattern P as its two CWs; and (2) x precedes y in P. Denote by $Nodes(G), Arcs(G)$ the nodes and arcs in a graph G, respectively.

We now compute three measures on $G(P)$ and combine them for all pattern candidates to filter asymmetric ones. The first measure (M_1) counts the proportion of words that can appear in both slots of the pattern, out of the total number of words. The reasoning here is that if a pattern allows a large percentage of words to participate in both slots, its chances of being a symmetric pattern are greater:

$$M_1 := \frac{|\{x|\exists y A(x,y) \wedge \exists z A(z,x)\}|}{|Nodes(G(P))|}$$

M_1 filters well patterns that connect words having different parts of speech. However, it may fail to filter patterns that contain multiple levels of asymmetric relationships. For example, in the pattern ‘x belongs to y’, we may find a word B on both sides (‘A belongs to B’, ‘B belongs to C’) while the pattern is still asymmetric.

In order to detect symmetric relationships in a finer manner, for the second and third measures we define $SymG(P)$, the symmetric subgraph of $G(P)$, containing only the bidirectional arcs and nodes of $G(P)$:

$$SymG(P) = \{\{x\}, \{(x,y)\}|A(x,y) \wedge A(y,x)\}$$

The second and third measures count the proportion of the number of symmetric nodes and edges in $G(P)$, respectively:

$$M_2 := \frac{|Nodes(SymG(P))|}{|Nodes(G(P))|}$$

[4] Considerations for the selection of thresholds are discussed in Section 5.

$$M_3 := \frac{|Arcs(SymG(P))|}{|Arcs(G(P))|}$$

All three measures yield values in $[0, 1]$, and in all three a higher value indicates more symmetry. M_2 and M_3 are obviously correlated, but they capture different aspects of a pattern's nature: M_3 is informative for highly interconnected but small word categories (e.g., month names), while M_2 is useful for larger categories that are more loosely connected in the corpus.

We use the three measures as follows. For each measure, we prepare a sorted list of all candidate patterns. We remove patterns that are not in the top Z_T (we use 100, see Section 5) in any of the three lists, and patterns that are in the bottom Z_B in at least one of the lists. The remaining patterns constitute our final list of symmetric patterns.

We do not rank the final list, since the category discovery algorithm of the next section does not need such a ranking. Defining and utilizing such a ranking is a subject for future work.

A sparse matrix representation of each graph can be computed in time linear in the size of the input corpus, since (1) the number of patterns $|P|$ is bounded, (2) vocabulary size $|V|$ (the total number of graph nodes) is much smaller than corpus size, and (3) the average node degree is much smaller than $|V|$ (in practice, with the thresholds used, it is a small constant.)

4 Discovery of Categories

After the end of the previous stage we have a set of symmetric patterns. We now use them in order to discover categories. In this section we describe the graph clique-set method for generating initial categories, and category pruning techniques for increased quality.

4.1 The Clique-Set Method

Our approach to category discovery is based on connectivity structures in the all-pattern word relationship graph G, resulting from merging all of the single-pattern graphs into a single unified graph. The graph G can be built in time $O(|V| \times |P| \times AverageDegree(G(P))) = O(|V|)$ (we use V rather than $Nodes(G)$ for brevity.)

When building G, no special treatment is done when one pattern is contained within another. For example, any pattern of the form CHC is contained in a pattern of the form HCHC ('x and y', 'both x and y'.) The shared part yields exactly the same subgraph. This policy could be changed for a discovery of finer relationships.

The main observation on G is that words that are highly interconnected are good candidates to form a category. This is the same general observation exploited by (Widdows and Dorow, 2002), who try to find graph regions that are more connected internally than externally.

We use a different algorithm. We find all strong n-cliques (subgraphs containing n nodes that are all bidirectionally interconnected.) A clique Q defines a category that contains the nodes in Q plus all of the nodes that are (1) at least unidirectionally connected to all nodes in Q, and (2) bidirectionally connected to at least one node in Q.

In practice we use 2-cliques. The strongly connected cliques are the bidirectional arcs in G and their nodes. For each such arc A, a category is generated that contains the nodes of all triangles that contain A and at least one additional bidirectional arc. For example, suppose the corpus contains the text fragments 'book and newspaper', 'newspaper and book', 'book and note', 'note and book' and 'note and newspaper'. In this case the three words are assigned to a category.

Note that a pair of nodes connected by a symmetric arc can appear in more than a single category. For example, suppose a graph G containing five nodes and seven arcs that define exactly three strongly connected triangles, ABC, ABD, ACE. The arc (A, B) yields a category $\{A, B, C, D\}$, and the arc (A, C) yields a category $\{A, C, B, E\}$. Nodes A and C appear in both categories. Category merging is described below.

This stage requires an $O(1)$ computation for each bidirectional arc of each node, so its complexity is $O(|V| \times AverageDegree(G)) = O(|V|)$.

4.2 Enhancing Category Quality: Category Merging and Corpus Windowing

In order to cover as many words as possible, we use the smallest clique, a single symmetric arc. This creates redundant categories. We enhance the quality of the categories by merging them and by windowing on the corpus.

We use two simple merge heuristics. First, if two categories are identical we treat them as one. Second, given two categories Q, R, we merge them iff there's more than a 50% overlap between them: $(|Q \bigcap R| > |Q|/2) \wedge (|Q \bigcap R| > |R|/2)$.

This could be added to the clique-set stage, but the phrasing above is simpler to explain and implement.

In order to increase category quality and remove categories that are too context-specific, we use a simple corpus windowing technique. Instead of running the algorithm of this section on the whole corpus, we divide the corpus into windows of equal size (see Section 5 for size determination) and perform the category discovery algorithm of this section on each window independently. Merging is also performed in each window separately. We now have a set of categories for each window. For the final set, we select only those categories that appear in at least two of the windows. This technique reduces noise at the potential cost of lowering coverage. However, the numbers of categories discovered and words they contain is still very large (see Section 5), so windowing achieves higher precision without hurting coverage in practice.

The complexity of the merge stage is $O(|V|)$ times the average number of categories per word times the average number of words per category. The latter two are small in practice, so complexity amounts to $O(|V|)$.

5 Evaluation

Lexical acquisition algorithms are notoriously hard to evaluate. We have attempted to be as thorough as possible, using several languages and both automatic and human evaluation. In the automatic part, we followed as closely as possible the methodology and data used in previous work, so that meaningful comparisons could be made.

5.1 Languages and Corpora

We performed in-depth evaluation on two languages, English and Russian, using three corpora, two for English and one for Russian. The first English corpus is the BNC, containing about 100M words. The second English corpus, Dmoz (Gabrilovich and Markovitch, 2005), is a web corpus obtained by crawling and cleaning the URLs in the Open Directory Project (dmoz.org), resulting in 68GB containing about 8.2G words from 50M web pages.

The Russian corpus was assembled from many web sites and carefully filtered for duplicates, to yield 33GB and 4G words. It is a varied corpus comprising literature, technical texts, news, newsgroups, etc.

As a preliminary sanity-check test we also applied our method to smaller corpora in Danish, Irish and Portuguese, and noted some substantial similarities in the discovered patterns. For example, in all 5 languages the pattern corresponding to 'x and y' was among the 50 selected.

5.2 Thresholds, Statistics and Examples

The thresholds T_H, T_C, T_P, Z_T, Z_B, were determined by memory size considerations: we computed thresholds that would give us the maximal number of words, while enabling the pattern access table to reside in main memory. The resulting numbers are $100, 50, 20, 100, 100$.

Corpus window size was determined by starting from a very small window size, defining at random a single window of that size, running the algorithm, and iterating this process with increased window sizes until reaching a desired vocabulary category participation percentage (i.e., x% of the different words in the corpus assigned into categories. We used 5%.) This process has only a negligible effect on running times, because each iteration is run only on a single window, not on the whole corpus.

The table below gives some statistics. V is the total number of different words in the corpus. W is the number of words belonging to at least one of our categories. C is the number of categories (after merging and windowing.) AS is the average category size. Running times are in minutes on a 2.53Ghz Pentium 4 XP machine with $1GB$ memory. Note how small they are, when compared to (Pantel et al, 2004), which took 4 days for a smaller corpus using the same CPU.

	V	W	C	AS	Time
Dmoz	16M	330K	142K	12.8	93m
BNC	337K	25K	9.6K	10.2	6.8m
Russian	10M	235K	115K	11.6	60m

Among the patterns discovered are the ubiquitous 'x and y', 'x or y' and many patterns containing them. Additional patterns include 'from x to y', 'x and/or y' (punctuation is treated here as white space), 'x and a y', and 'neither x nor y'.

We discover categories of different parts of speech. Among the noun ones, there are many whose precision is 100%: 37 countries, 18 languages, 51 chemical elements, 62 animals, 28 types of meat, 19 fruits, 32 university names, etc. A nice verb category example is {*dive, snorkel, swim, float, surf, sail, canoe, kayak, paddle, tube, drift*}. A nice adjective example is {*amazing,* *awesome, fascinating, inspiring, inspirational, exciting, fantastic, breathtaking, gorgeous.*}

5.3 Human Judgment Evaluation

The purpose of the human evaluation was dual: to assess the quality of the discovered categories in terms of precision, and to compare with those obtained by a baseline clustering algorithm.

For the baseline, we implemented k-means as follows. We have removed stopwords from the corpus, and then used as features the words which appear before or after the target word. In the calculation of feature values and inter-vector distances, and in the removal of less informative features, we have strictly followed (Pantel and Lin, 2002). We ran the algorithm 10 times using $k = 500$ with randomly selected centroids, producing 5000 clusters. We then merged the resulting clusters using the same 50% overlap criterion as in our algorithm. The result included 3090, 2116, and 3206 clusters for Dmoz, BNC and Russian respectively.

We used 8 subjects for evaluation of the English categories and 15 subjects for evaluation of the Russian ones. In order to assess the subjects' reliability, we also included random categories (see below.)

The experiment contained two parts. In Part I, subjects were given 40 triplets of words and were asked to rank them using the following scale: (1) the words definitely share a significant part of their meaning; (2) the words have a shared meaning but only in some context; (3) the words have a shared meaning only under a very unusual context/situation; (4) the words do not share any meaning; (5) I am not familiar enough with some/all of the words.

The 40 triplets were obtained as follows. 20 of our categories were selected at random from the non-overlapping categories we have discovered, and three words were selected from each of these at random. 10 triplets were selected in the same manner from the categories produced by k-means, and 10 triplets were generated by random selection of content words from the same window in the corpus.

In Part II, subjects were given the full categories of the triplets that were graded as 1 or 2 in Part I (that is, the full 'good' categories in terms of sharing of meaning.) They were asked to grade the categories from 1 (worst) to 10 (best) according to how much the full category had met the expectations they had when seeing only the triplet.

Results are given in Table 1. The first line gives the average percentage of triplets that were given scores of 1 or 2 (that is, 'significant shared meaning'.) The 2nd line gives the average score of a triplet (1 is best.) In these lines scores of 5 were not counted. The 3rd line gives the average score given to a full category (10 is best.) Inter-evaluator Kappa between scores 1,2 and 3,4 was 0.56, 0.67 and 0.72 for Dmoz, BNC and Russian respectively.

Our algorithm clearly outperforms k-means, which outperforms random. We believe that the Russian results are better because the percentage of native speakers among our subjects for Russian was larger than that for English.

5.4 WordNet-Based Evaluation

The major guideline in this part of the evaluation was to compare our results with previous work having a similar goal (Widdows and Dorow, 2002). We have followed their methodology as best as we could, using the same WordNet (WN) categories and the same corpus (BNC) in addition to the Dmoz and Russian corpora[5].

The evaluation method is as follows. We took the exact 10 WN subsets referred to as 'subjects' in (Widdows and Dorow, 2002), and removed all multi-word items. We now selected at random 10 pairs of words from each subject. For each pair, we found the largest of our discovered categories containing it (if there isn't one, we pick another pair. This is valid because our Recall is obviously not even close to 100%, so if we did not pick another pair we would seriously harm the validity of the evaluation.) The various morphological forms of the same word were treated as one during the evaluation.

The only difference from the (Widdows and Dorow, 2002) experiment is the usage of pairs rather than single words. We did this in order to disambiguate our categories. This was not needed in (Widdows and Dorow, 2002) because they had directly accessed the word graph, which may be an advantage in some applications.

The Russian evaluation posed a bit of a problem because the Russian WordNet is not readily available and its coverage is rather small. Fortunately, the subject list is such that WordNet words

[5] (Widdows and Dorow, 2002) also reports results for an LSA-based clustering algorithm that are vastly inferior to the pattern-based ones.

	Dmoz			BNC			Russian		
	us	k-means	random	us	k-means	random	us	k-means	random
avg 'shared meaning' (%)	**80.53**	18.25	1.43	**86.87**	8.52	0.00	**95.00**	18.96	7.33
avg triplet score (1-4)	**1.74**	3.34	3.88	**1.56**	3.61	3.94	**1.34**	3.32	3.76
avg category score (1-10)	**9.27**	4.00	1.8	**9.31**	4.50	0.00	**8.50**	4.66	3.32

Table 1: Results of evaluation by human judgment of three data sets (ours, that obtained by k-means, and random categories) on the three corpora. See text for detailed explanations.

could be translated unambiguously to Russian and words in our discovered categories could be translated unambiguously into English. This was the methodology taken.

For each found category C containing N words, we computed the following (see Table 2): (1) Precision: the number of words present in both C and WN divided by N; (2) Precision*: the number of correct words divided by N. Correct words are either words that appear in the WN subtree, or words whose entry in the American Heritage Dictionary or the Britannica directly defines them as belonging to the given class (e.g., 'keyboard' is defined as 'a piano'; 'mitt' is defined by 'a type of glove'.) This was done in order to overcome the relative poorness of WordNet; (3) Recall: the number of words present in both C and WN divided by the number of (single) words in WN; (4) The number of correctly discovered words (New) that are not in WN. The Table also shows the number of WN words (:WN), in order to get a feeling by how much WN could be improved here. For each subject, we show the average over the 10 randomly selected pairs.

Table 2 also shows the average of each measure over the subjects, and the two precision measures when computed on the total set of WN words. The (uncorrected) precision is the only metric given in (Widdows and Dorow, 2002), who reported 82% (for the BNC.) Our method gives 90.47% for this metric on the same corpus.

5.5 Summary

Our human-evaluated and WordNet-based results are better than the baseline and previous work respectively. Both are also of good standalone quality. Clearly, evaluation methodology for lexical acquisition tasks should be improved, which is an interesting research direction in itself.

Examining our categories at random, we found a nice example that shows how difficult it is to evaluate the task and how useful automatic category discovery can be, as opposed to manual definition. Consider the following category, discovered in the Dmoz corpus: $\{$*nightcrawlers, chicken, shrimp, liver, leeches*$\}$. We did not know why these words were grouped together; if asked in an evaluation, we would give the category a very low score. However, after some web search, we found that this is a 'fish bait' category, especially suitable for catfish.

6 Discussion

We have presented a novel method for pattern-based discovery of lexical semantic categories. It is the first pattern-based lexical acquisition method that is fully unsupervised, requiring no corpus annotation or manually provided patterns or words. Pattern candidates are discovered using meta-patterns of high frequency and content words, and symmetric patterns are discovered using simple graph-theoretic measures. Categories are generated using a novel graph clique-set algorithm. The only other fully unsupervised lexical category acquisition approach is based on decomposition of a matrix defined by context feature vectors, and it has not been shown to scale well yet. Our algorithm was evaluated using both human judgment and automatic comparisons with WordNet, and results were superior to previous work (although it used a POS tagged corpus) and more efficient computationally. Our algorithm is also easy to implement.

Computational efficiency and specifically lack of annotation are important criteria, because they allow usage of huge corpora, which are presently becoming available and growing in size.

There are many directions to pursue in the future: (1) support multi-word lexical items; (2) increase category quality by improved merge algorithms; (3) discover various relationships (e.g., hyponymy) between the discovered categories; (4) discover finer inter-word relationships, such as verb selection preferences; (5) study various properties of discovered patterns in a detailed manner; and (6) adapt the algorithm to morphologically rich languages.

Subject	Prec.	Prec.*	Rec.	New:WN
Dmoz				
instruments	79.25	89.34	34.54	7.2:163
vehicles	80.17	86.84	18.35	6.3:407
academic	78.78	89.32	30.83	15.5:396
body parts	73.85	79.29	5.95	9.1:1491
foodstuff	83.94	90.51	28.41	26.3:1209
clothes	83.41	89.43	10.65	4.5:539
tools	83.99	89.91	21.69	4.3:219
places	76.96	84.45	25.82	6.3:232
crimes	76.32	86.99	31.86	4.7:102
diseases	81.33	88.99	19.58	6.8:332
set avg	79.80	87.51	22.77	9.1:509
all words	79.32	86.94		
BNC				
instruments	92.68	95.43	9.51	0.6:163
vehicles	94.16	95.23	3.81	0.2:407
academic	93.45	96.10	12.02	0.6:396
body parts	96.38	97.60	0.97	0.3:1491
foodstuff	93.76	94.36	3.60	0.6:1209
cloths	93.49	94.90	4.04	0.3:539
tools	96.84	97.24	6.67	0.1:219
places	87.88	97.25	6.42	1.5:232
crimes	83.79	91.99	19.61	2.6:102
diseases	95.16	97.14	5.54	0.5:332
set avg	92.76	95.72	7.22	0.73:509
all words	90.47	93.80		
Russian				
instruments	82.46	89.09	25.28	3.4:163
vehicles	83.16	89.58	16.31	5.1:407
academic	87.27	92.92	15.71	4.9:396
body parts	81.42	89.68	3.94	8.3:1491
foodstuff	80.34	89.23	13.41	24.3:1209
clothes	82.47	87.75	15.94	5.1:539
tools	79.69	86.98	21.14	3.7:219
places	82.25	90.20	33.66	8.5:232
crimes	84.77	93.26	34.22	3.3:102
diseases	80.11	87.70	20.69	7.7:332
set avg	82.39	89.64	20.03	7.43:509
all words	80.67	89.17		

Table 2: WordNet evaluation. Note the BNC ‘all words’ precision of 90.47%. This metric was reported to be 82% in (Widdows and Dorow, 2002).

It should be noted that our algorithm can be viewed as one for automatic discovery of word senses, because it allows a word to participate in more than a single category. When merged properly, the different categories containing a word can be viewed as the set of its senses. We are planning an evaluation according to this measure after improving the merge stage.

References

Matthew Berland and Eugene Charniak, 1999. Finding parts in very large corpora. ACL ’99.

Peter Brown, Vincent Della Pietra, Peter deSouza, Jenifer Lai, Robert Mercer, 1992. Class-based n-gram models for natural language. *Comp. Linguistics*, 18(4):468–479.

Sharon Caraballo, 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. ACL ’99.

Timothy Chklovski, Patrick Pantel, 2004. VerbOcean: mining the web for fine-grained semantic verb relations. EMNLP ’04.

Philipp Cimiano, Andreas Hotho, Steffen Staab, 2005. Learning concept hierarchies from text corpora using formal concept analysis. *J. of Artificial Intelligence Research*, 24:305–339.

James Curran, Marc Moens, 2002. Improvements in automatic thesaurus extraction. ACL Workshop on Unsupervised Lexical Acquisition, 2002.

Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, Richard Harshman, 1990. Indexing by latent semantic analysis. *J. of the American Society for Info. Science*, 41(6):391–407.

Beate Dorow, Dominic Widdows, Katarina Ling, Jean-Pierre Eckmann, Danilo Sergi, Elisha Moses, 2005. Using curvature and Markov clustering in graphs for lexical acquisition and word sense discrimination. MEANING ’05.

Dayne Freitag, 2004. Trained named entity recognition using distributional clusters. EMNLP ’04.

Evgeniy Gabrilovich, Shaul Markovitch, 2005. Feature generation for text categorization using world knowledge. IJCAI ’05.

Marti Hearst, 1992. Automatic acquisition of hyponyms from large text corpora. COLING ’92.

Hang Li, Naoki Abe, 1996. Clustering words with the MDL principle. COLING ’96.

Dekang Lin, 1998. Automatic retrieval and clustering of similar words. COLING ’98.

Margaret Matlin, 2005. *Cognition, 6th edition.* John Wiley & Sons.

Patrick Pantel, Dekang Lin, 2002. Discovering word senses from text. SIGKDD ’02.

Patrick Pantel, Deepak Ravichandran, Eduard Hovy, 2004. Towards terascale knowledge acquisition. COLING ’04.

Fernando Pereira, Naftali Tishby, Lillian Lee, 1993. Distributional clustering of English words. ACL ’93.

Ellen Riloff, Rosie Jones, 1999. Learning dictionaries for information extraction by multi-level bootstrapping. AAAI ’99.

Hinrich Schütze, 1998. Automatic word sense discrimination. *Comp. Linguistics,* 24(1):97–123.

Dominic Widdows, Beate Dorow, 2002. A graph model for unsupervised Lexical acquisition. COLING ’02.