Customizing a Lexicon to Better Suit a Computational Task

Marti A. Hearst

Computer Science Division
571 Evans Hall, UC Berkeley
Berkeley, CA 94720 USA
marti@cs.berkeley.edu

Hinrich Schütze

Linguistics Department
Stanford University
Stanford, CA 94305 USA
schuetze@csli.stanford.edu

Abstract

We discuss a method for augmenting and rearranging a structured lexicon in order to make it more suitable for a topic labeling task, by making use of lexical association information from a large text corpus. We first describe an algorithm for converting the hierarchical structure of WordNet [13] into a set of flat categories. We then use lexical cooccurrence statistics in combination with these categories to classify proper names, assign more specific senses to broadly defined terms, and classify new words into existing categories. We also describe how to use these statistics to arrange categories into schema-like groupings and show how the new categories improve a text-labeling algorithm. In effect, we provide a mechanism for successfully combining a hand-built lexicon with knowledge-free, statistically-derived information.

Introduction

Much effort is being applied to the creation of lexicons and the acquisition of semantic and syntactic attributes of the lexical items that comprise them, e.g., [1], [4], [7], [8], [11], [16], [19]. However, a lexicon as given may not suit the requirements of a particular computational task. Because lexicons are expensive to build, rather than create a new one from scratch, it can preferable to adjust an existing one to meet an application's needs. In this chapter we describe such an effort: we add associational information to a hierarchically structured lexicon in order to better serve a text labeling task.

For the purposes of information retrieval, we would like to assign labels to full-text documents that characterize their main topics.[1] These labels

[1] The terms "label", "topic", and "category" are used interchangeably in this chapter.

are to be used in an interface that displays the texts that have been retrieved in response to a query according to intersections of the texts' main topics (see [9]).

One way to label texts, when working within a limited domain of discourse, is to start with a pre-defined set of topics and specify the word contexts that indicate the topics of interest (e.g., [10]). Another way, assuming that a large collection of pre-labeled texts exists, is to use statistics to automatically infer which lexical items indicate which labels (e.g., [12]). In contrast, we are interested in assigning labels to general, domain-independent text, without benefit of pre-classified texts. In all three cases, a lexicon that specifies which lexical items correspond to which topics is required. The topic labeling method we use is statistical and thus requires a large number of representative lexical items for each category.

The starting point for our lexicon is WordNet [13], which is readily available online and provides a large repository of English lexical items. WordNet[2] is composed of *synsets*, structures containing sets of terms with synonymous meanings, thus allowing a distinction to be made between different senses of homographs. Associated with each synset is a list of relations that the synset participates in. One of these, in the noun dataset, is the hyponymy relation (and its inverse, hypernymy), roughly glossed as the "ISA" relation. This relation imposes a hierarchical structure on the synsets, indicating how to generalize from a subordinate term to a superordinate one, and vice versa.[3] This is a very useful kind of information for many tasks, such as word class generalization [17]. Figure 0.1 shows an example of one small piece of the lexicon.

We would like to adjust this lexicon in two ways in order to facilitate the label assignment task. The first is to collapse the fine-grained hierarchical structure into a set of coarse but semantically-related categories. These categories will provide the lexical evidence for the topic labels. (After the label is assigned, the hierarchical structure can be reintroduced.) Once the hierarchy has been converted into categories, we can augment the categories with new lexical items culled from free text corpora, in order to further improve the labeling task.

[2] All work described here pertains to Version 1.3 of WordNet.

[3] Actually, the hyponomy relation is a directed acyclic graph, in that a minority of the nodes are children of more than one parent. We will at times refer to it as a hierarchy nonetheless.

```
2 senses of bread

Sense 1
shekels, gelt, dough, bread, dinero, lucre, loot,
      pelf, moolah, cabbage
       => money
           => tender, legal tender
               => medium of exchange, monetary system
                   => asset
                       => possession

Sense 2
bread, breadstuff, staff of life
       => baked good, baked goods
           => aliment, nourishment, nutriment,
              sustenance, victuals
               => food, nutrient
                   => substance, matter
                       => object, inanimate object,
                          physical object, thing
                           => entity
```

Figure 0.1
The hypernyms (ancestors) of *bread* in WordNet (Version 1.3).

The second way we would like to adjust the lexicon is to combine categories from distant parts of the hierarchy. In particular, we are interested in finding groupings of terms that contribute to a frame or schema-like representation [14]; this can be achieved by finding associational lexical relations among the existing taxonymic relations. For example, WordNet has the following synsets: "athletic game" (hyponyms: baseball, tennis), "sports implement" (hyponyms: bat, racquet), and "tract, piece of land" (hyponyms: baseball_diamond, court), none of which are closely related in the hierarchy. We would like to automatically find relations among categories headed by synsets like these. (In Version 1.3, the WordNet encoders have placed some associational links among these categories, but still only some of the desired connections appear.)

In other words, we would like to derive links among schematically related parts of the hierarchy, where these links reflect the text genre on which text processing is to be done. [18] describes a method called WordSpace that represents lexical items according to how semantically close they are to one another, based on evidence from a large text corpus. We propose combining this term-similarity information with the hierarchical information already available in WordNet to create structured associational information.

In the next section we describe the algorithm for compressing the WordNet hierarchy into a set of categories. This is followed by a discussion of how these categories are to be used and why they need to be improved. This is followed by a discussion of the first improvement technique: including new, related terms from a corpus. The next section describes the second improvement technique: bringing disparate categories together to form schematic groupings while retaining the given hierarchical structure. The final section concludes the chapter.

Creating Categories from WordNet

We would like to decompose the WordNet noun hierarchy into a set of disjoint categories, each consisting of a relatively large number of synsets. (This is necessary for the text-labeling task, because each topic must be represented by many different terms.) The goal of creating categories of a particular average size with as small a variance as possible. There is some limit as to how small this variance can be because there are several

synsets that have a very large number of children (there are sixteen nodes with a branching factor greater than 100). This primarily occurs with synsets of a taxonymic flavor, i.e., mushroom species and languages of the world. There are two other reasons why it is not straightforward to find uniformly sized, meaningful categories:

(i) There is no explicit measure of semantic distance among the children of a synset.

(ii) The hierarchy is not balanced, i.e., the depth from root to leaf varies dramatically throughout the hierarchy, as does the branching factor. (The hierarchy has ten root nodes; on average their maximum depth is 10.5 and their minimum depth is 2.)

Reason (ii) rules out a strategy of traveling down a uniform depth from the root or up a uniform height from the leaves in order to achieve uniform category sizes.

For the purposes of the description of this algorithm, a synset is a node in the hierarchy, and a descendant of synset N is any synset reachable via a hyponym link from N or any of N's descendants (recursively). This means that intermediate, or non-leaf synsets, are also classified as descendants. The term "child" refers to an immediate descendant, i.e., a synset directly linked to N via a hyponym link.

The algorithm used here is controlled by two parameters: upper and lower bounds on the category size (see Figure 0.2). For example, the result of setting the lower bound to 25 and the upper bound to 60 yields categories with an average size of 58 members. An arbitrary node N in the hierarchy is chosen, and if it has not yet been marked as a member of a category, the algorithm checks to see how many unmarked descendants it has. In every case, if the number of descendants is too small, the assignment to a category is deferred until a node higher in the hierarchy is examined (unless the node has no parents). This helps avoid extremely small categories, which are especially undesirable.

If the number of descendants of N falls within the boundaries, the node and its unmarked descendants are bundled into a new category, marked, and assigned a label which is derived from the synset at N. Thus, if N and its unmarked descendants create a category with k members, the number of unmarked descendants of the parent of N decreases by k.

```
for each synset N in the noun hierarchy
  a_cat(N)

  a_cat(N):
  if N is unmarked (has not been entered in a category)
      T <- #descendents(N)

      if ((T >= LOWER_BRACKET)
            && (T <= UPPER_BRACKET))
         mark(N,NewCatNumber)

      else if (T > UPPER_BRACKET)

        for each (direct) child C of N
           CT <- #descendents(C)
           if ((CT >= LOWER_BRACKET)
               && (CT <= UPPER_BRACKET))
              mark(C,NewCatNumber)
           else if (CT > UPPER_BRACKET)
               a_cat(C)

        T <- #descendents(N)
        if (T >= LOWER_BRACKET)
           mark(N,NewCatNumber)
```

Figure 0.2
Algorithm for creating categories from WordNet's noun hierarchy.

If N has too many descendants, that is, the count of its unmarked descendants exceeds the upper bound, then each of its immediate children is checked in turn: if the child's descendant count falls between the boundaries, then the child and its descendants are bundled into a category. If the child and its unmarked descendants exceed the upper bound, then the procedure is called recursively on the child. Otherwise, the child is too small and is left alone. After all of N's children have been processed, the category that N will participate in has been made as small as the algorithm will allow. There is a chance that N and its unmarked descendants will now make a category that is too small, and if this is the case, N is left alone, and a higher-up node will eventually subsume it (unless N has no parents remaining). Otherwise, N and its remaining unmarked descendants are bundled into a category.

If N has more than one parent, N can end up assigned to the category of any of its parents (or none), depending on which parent was accessed first and how many unmarked children it had at any time, but each synset is assigned to only one category.

The function "mark" places the synset and all its descendents that have not yet been entered into a category into a new category. Note that #descendents is recalculated in the third-to-last line in case any of the children of N have been entered into categories.

In the end there may be isolated small pieces of hierarchy that are not stored in any category, but this can be fixed by a cleanup pass, if desired.

A Topic Labeler

We are using a version of the disambiguation algorithm described in [20] to assign topic labels to coherent passages of text. Yarowsky defines word senses as the categories listed for a word in *Roget's Thesaurus* (Fourth Edition), where a category is something like TOOLS/MACHINERY. For each category, the algorithm

- Collects contexts that are representative of the category.
- Identifies salient words in the collective contexts and determines the weight for each word.
- Uses the resulting weights to predict the appropriate category for a word occurring in a novel context.

The proper use of this algorithm is to choose among the categories to which a particular ambiguous word can belong, based on the lexical context that surrounds a particular instance of the word.

In our implementation of the algorithm, the 726 categories derived from WordNet, as described in the previous section, are used instead of *Roget's* categories, because these are not available publicly online. Training is performed on *Grolier's American Academic Encyclopedia* ($\approx$ 8.7M words).

The labeling is done as follows: instead of using the algorithm in the intended way, we are placing probes in the text at evenly-spaced intervals and accumulating the scores for each category all the way through the text. The intention is that at the end the highest scoring categories

	United States Constitution	*Genesis*
0	assembly (court, legislature)	deity divinity god
1	due_process_of_law	relative relation (mother, aunt)
2	legal_document	worship
3	administrative_unit	man adult_male
4	body (legislative)	professional
5	charge (taxes)	happiness gladness felicity
6	administrator decision_maker	woman adult_female
7	document written_document	evildoing transgression
8	approval (sanction, pass)	literary_composition
9	power powerfulness	religionist religious_person

Figure 0.3
Output using original category set on two well-known texts.

correspond to the main topics of the text. This algorithm is described in more detail in [9]. Figure 0.3 shows the output of the labeler on two well-known texts (made available online by Project Gutenberg). The first column indicates the rank of the category, and the second and third columns show the words in the synset at the top-most node of the category for each text (these are not always entirely descriptive, so some glosses are provided in parentheses).

Note that although most of the categories are appropriate (with the glaring exception of "professional" in *Genesis*), there is some redundancy among them, and in some cases they are too fine-level to indicate main topic information.

In an earlier implementation of this algorithm, the categories were in general larger but less coherent than in the current set. The larger categories resulted in better-trained classifications, but the classes often conflated quite disparate terms. The current implementation produces smaller, more coherent categories. The advantage is that a more distinct meaning can be associated with a particular label, but the disadvantage is that in many cases so few of the words in the category appear in the training data that a weak model is formed. Then the categories with little distinguishing training data dominate the labeling scores inappropriately.

In the category-derivation algorithm described above, in order to increase the size of a given category, terms must be taken from nodes

adjacent in the hierarchy (either descendants or siblings). However, adjacent terms are not necessarily closely related semantically, and so after a point, expanding the category via adjacent terms introduces noise. To remedy this problem, we have experimented with increasing the size of the categories in two different ways:

(1) The first approach is to retain the categories in their current form and add semantically similar terms, extracted from corpora independent of WordNet, thus improving the training of the labeling algorithm.

(2) The second approach is to determine which categories are semantically related to one another, despite the fact that they come from quite different parts of the hierarchy, and combine them so that they form schema-like associations.

These are described in the next two sections, respectively.

Augmenting Categories with Relevant Terms

As mentioned above, one way to improve the categories is to expand them with related relevant terms. In this section we show how comparing WordSpace vectors to the derived categories allows us to expand the categories. The first subsection describes the WordSpace algorithm, and the subsequent subsections show how it can be used to augment the derived categories.

Creating WordSpace from Free Text

WordSpace [18] is a corpus-based method for inducing semantic representations for a large number of words (50,000) from lexical cooccurrence statistics. The representations are derived from free text, and therefore are highly specific to the text type in question. The medium of representation is a multi-dimensional, real-valued vector space. The cosine of the angle between two vectors in the space is a continuous measure of their semantic relatedness.

Lexical coocurrence, which is the basis for creating the word space vectors, can be easily measured. However, for a vocabulary of 50,000 words, there are 2,500,000,000 possible coocurrence counts, a number too high to be computationally tractable. Therefore, *letter fourgrams* are used here to bootstrap the representations. Cooccurrence statistics

are collected for 5,000 selected fourgrams. The 5000-by-5000 matrix used for this purpose is manageable. A vector for a lexical item is then computed as the sum of fourgram vectors that occur close to it in the text.

The first step of the creation of WordSpace consists of deriving fourgram vectors that reflect semantic similarity in the sense of being used to describe the same contexts. Consequently, one needs to be able to pairwise compare fourgrams' contexts. For this purpose, a *collocation matrix* for fourgrams was collected such that the entry $a_{i,j}$ counts the number of times that fourgram i occurs at most 200 fourgrams to the left of fourgram j. Two columns in this matrix are similar according to the cosine measure if the contexts the corresponding fourgrams are used in are similar. The counts were determined using five months of the New York Times (June – October 1990). The resulting collocation matrix is dense: only 2% of entries are zeros, because almost any two fourgrams cooccur. Only 10% of entries are smaller than 10, so that culling small counts would not increase the sparseness of the matrix. Consequently, any computation that employs the fourgram vectors directly would be inefficient. For this reason, a singular value decomposition was performed and 97 singular values extracted (cf. [5]) using an algorithm from SVDPACK [3]. Each fourgram can then be represented by a vector of 97 real values. Since the singular value decomposition finds the best least-square approximation of the original space in 97 dimensions, two fourgram vectors will be similar if their original vectors in the collocation matrix are similar. The reduced fourgram vectors can be efficiently used in the following computations.

Cooccurrence information was used for a second time to compute word representations from the fourgram vectors: in this case coocurrence of a target word with any of the 5000 fourgrams. 50,000 words that occurred at least 20 times in 50,000,000 words of the New York Times newswire were selected. For each of the words, a context vector was computed for every position at which it occurred in the text. A context vector was defined as the sum of all defined fourgram vectors in a window of 1001 fourgrams centered around the target word. The context vectors were then normalized and summed. This sum of vectors is the vector representation of the target word. It is the *confusion* of all its uses in the corpus. More formally, if $C(w)$ is the set of positions in the corpus at which w occurs and if $\varphi(f)$ is the vector representation for fourgram f,

word	nearest neighbors
burglar	burglars thief rob mugging stray robbing lookout chase
disable	deter intercept repel halting surveillance shield maneuvers
disenchantment	disenchanted sentiment resentment grudging mindful
domestically	domestic auto/-s importers/-ed threefold inventories
Dour	melodies/-dic Jazzie danceable reggae synthesizers Soul
grunts	heap into ragged goose neatly pulls buzzing rake odd
kid	dad kidding mom ok buddies Mom Oh Hey hey mama
S.O.B.	Confessions Jill Julie biography Judith Novak Lois
Ste.	dry oyster whisky hot filling rolls lean float bottle ice
workforce	jobs employ/-s/-ed/-ing attrition workers clerical

Table 0.1
Ten random words and their nearest neighbors.

then the vector representation $\tau(w)$ of w is defined as: (the dot stands for normalization)

$$\tau(w) = \sum_{i \in C(w)} (\sum_{f \text{ close to } i}^{\bullet} \varphi(f))$$

Table 0.1 shows a random sample of 10 words and their nearest neighbors in Word Space (using the cosine measure). As can be seen from the table, proximity in the space corresponds closely to semantic similarity in the corpus. (*N'Dour* is a Senegalese jazz musician. In the 1989/90 New York Times, *S.O.B.* mainly occurs in the book title "Confessions of an S.O.B.", and *Ste.* in the name "Ste.-Marguerite" a Quebec river that is popular for salmon fishing.)

Augmenting WordNet Categories using WordSpace

We chose the following simple mapping from the derived WordNet categories to WordSpace:

- for each word w in WordSpace
- collect the 20 nearest neighbors of w in the space
- compute the score s_i of category i for w as the number of nearest neighbors that are in i
- assign w to the highest scoring category or categories

In order to test this algorithm, we selected 1000 words from the medium frequency words in WordSpace.[4] These turned out to be the medium-frequency words from *deforestation* to *downed.* The following subsections describe the application of the assignment algorithm to classifying proper names, reassigning words in the categories, and assigning words that are not covered by the categories.

Semantic classification of proper names A deficiency of WordNet for our text labeling task and for many other applications is that it omits many proper names (and since the set of important proper names changes over time, it cannot be expected to contain an exhaustive list). We tested the performance of our assignment algorithm by searching for proper names that had high scores for the categories in Table 0.2. For each category on the left-hand side we show all of the proper names that assigned high scores to those categories. The proper names assigned to "artist" are painters, the proper names assigned to "European country" are European politicians, "performer" contains actors, dancers and roles, writers and titles of movies, "music" has musicians and titles of musical performances (the Pasadena Doo Dah Parade, Purcell's "Dido and Aeneas"), "athlete jock" players of various sports, and "process of law" lawyers, judges and defendants. We checked the referents of all proper names in Table 0.2 in the New York Times and found only one error: The President of Michigan State University, John DiBiaggio, was assigned to the "athlete" category because his name is mainly mentioned in articles dealing with a conflict he had with his athletic department. However, a few names like "DePalma" and "Delancey" had several referents only one of whom pertained to the assigned category.

Fine-tuning WordNet terms The assignment algorithm can also be employed to adjust the assignments of individual words in the WordNet hierarchy by matching against the derived categories. Two kinds of adjustments are possible: specializing senses and adding senses that are not covered. Two examples of each case from the 1000 word test set are given in Table 0.3.

[4] WordSpace has three parts: high-frequency, medium-frequency, and low-frequency words. The words in the test set have the internal identification numbers 26,000 through 26,999.

category	highest scoring proper names
artist creative_person	degas delacroix
European_country European_nation	delors dienstbier diestel
performer performing_artist; dramatic composition	deniro dennehy depalma delancey depardieu dern desi devito dewhurst dey diaghilev doogie dourif
musical_organization musical_group; musician player; music	depeche(mode) deville diddley dido dire(straits) doo doobie (n')dour
athlete jock	dehere delpino demarco deleon deshaies detmer dibiaggio dinah doleman doughty doran dowis
due_process due_process_of_law	degeorge depetris devita dichiara dicicco diles dilorenzo dougan

Table 0.2
Assigning proper names to WordNet categories.

word	highest scoring category
dosage	medicine medication medicament
dissertation	science scientific_discipline
Derbies	horse Equus_caballus
dl	athlete jock

Table 0.3
Finding terms that characterize the senses of words better than the corresponding WordNet assignment does (see Table 0.4).

word	WordNet definition
dosage	dose, dosage – (the quantity of an active agent (substance or radiation) taken in or absorbed at any one time)
dissertation	dissertation, thesis => treatise – (a formal exposition)
derby	bowler hat, bowler, derby, plug hat – (round and black and hard with a narrow brim; worn by some British businessmen)
dl	deciliter, decilitre, dl

Table 0.4
Synonym sets in WordNet for the words in Table 0.3.

word	score	highest scoring categories
degradable	+	compound chemical_compound
demagoguery	0	feeling emotion
deprenyl	+	infectious_disease; disease
desktop	+	memory_device storage_device
deuterium	+	chemical_element element; substance matter
(pas de) deux	+	dancing dance terpsichore
dideoxyinosine	+	medicine medication medicament; infectious_disease
(per) diem	+	commercial_document/instrument; occupation
dieters	+	foodstuff
dinnerware	+	tableware
dioxins	+	chemical_element element
dispersants	0	change alteration modification
disservice	−	cognitive_state state_of_mind
dissidence	+	leader; social_group
disunity	−	speech_act
diuretic	+	symptom
diuretics	+	disease; liquid_body_substance body_fluid
doctrinal	+	religion faith church
dogfight	+	happening occurrence; conflict struggle
doggie	−	unpleasant_person persona_non_grata
doggone	−	unit_of_measurement unit; integer whole_number
Domaine	+	wine vino
domesticity	−	person individual man mortal human soul; feeling
dopamine	+	medicine medication medicament; room
dossier	0	statement; message content subject_matter substance
doubleheaders	0	time_period period period_of_time amount_of_time
downbeat	0	message content subject_matter substance; feeling

Table 0.5
Assigning unknown words to semantically relevant categories. Key: + correct; 0 related; - incorrect.

dosage and *dissertation* are defined in a very general way in Wordnet (see Table 0.4). While they can be used with the general sense given in WordNet, almost all uses of *dissertation* in the New York Times are for doctoral dissertations that report on scientific work. Similarly, non-medical contexts are conceivable for *dosage*, but the dosages that the New York Times mentions are exclusively dosages of radiation or medicine in a medical context. The automatically found labelings in Table 0.3 indicate the need for specialization and can be used as the basis for reassignment.

In some cases, the WordNet hierarchy is also incomplete. The two senses "horse race" and "Disabled List" for *derby* and *dl* are missing from WordNet, although they are the dominant uses in the New York Times. Again the classification algorithm finds the right topic area for the two words which can be used as the basis for reassignment. Unfortunately, the algorithm also labels some correctly assigned words with incorrect categories.

Assigning unknown words We would like to be able to handle unknown words since they are often highly specific and excellent indicators for the topical structure of a document. Table 0.5 shows the automatic assignments for all words in the 1000 word test set that were not found in WordNet.

The results are mixed. 63% (17/27) of the words are assigned to a correct topic (+), an additional 19% (5/27) are assigned to topics they are related to (0), 19% are misassigned (–). We are considering several ways of improving the assignment algorithm. For instance, there are "diluted" categories such as "speech_act" and "trait character feature" whose members are mostly words that are poorly characterized collocationally. If we ignore them in assigning categories (hoping that most unknown words will be topic-specific special terms) we can correct some of the errors, e.g., *disunity* would be assigned to "group_action interaction social_activity" which seems correct. We expect that we can improve the results in Table 0.5 as we gain more experience in combining WordSpace and WordNet.

These results are encouraging; we have not yet tested to see if they improve the topic labeler, however.

Combining Distant Categories

This section describes an algorithm that uses Word Space vectors to help determine which WordNet-derived categories should be combined to create new, more schema-like categories.

The Algorithm

To find which categories should be considered closest to one another, we first determined how close they are in WordSpace and then grouped categories together that mutually ranked one another highly.

To compute the first-degree closeness of two categories c_i and c_j we used the formula:

$$D(c_i, c_j) = \frac{1}{2}\frac{1}{|c_i||c_j|}\sum_{\vec{v}\in c_i}\sum_{\vec{w}\in c_j} d(\vec{v}, \vec{w})$$

where d is the square of the Euclidean distance:

$$d(\vec{v}, \vec{w}) = \sum_i (v_i - w_i)^2$$

The primary rank of category i for category j indicates how closely related i is to j. For instance rank 1 means that i is the closest category to j, and rank 3 means there are only two closer categories to j than i.

The second-degree closeness is computed from the rank of the primary ranks. To determine that close association is mutual between two categories, we check for mutual high ranking. Thus category i and j are grouped together if and only if i ranks j highly and j ranks i highly (where "highly" was determined by a cutoff value – i and j had to be ranked k or above with respect to each other, for a threshold k). Secondary ranking is needed because some categories are especially "popular," attracting many other categories to them; the secondary rank enables the popular categories to retain only those categories that they mutually rank highly.

The results of this algorithm are best interpreted via a graphical layout. Figure 0.4 shows a piece of a network created using a presentation tool [2] based on theoretical work by [6]. The underlying algorithm uses a force-directed placement model to layout complex networks (edges are modeled as springs; nodes linked by edges are attracted to each other, but all other pairs of nodes are repelled from one another).

In these networks only connectivity has meaning; distance between nodes does not connote semantic distance. The connectivity of the network is interesting also because it indicates the interconnectivity between categories. From Figure 0.4, we see that categories associated with the notion *sports*, such as *athletic_game, race, sports_equipment,* and *sports_implement,* have been grouped together. *Athletics* is linked to *vehicle* and *competition* categories; these in turn link to *military_vehicles* and *weaponry* categories, which then lead in to *legal* categories.

The network also shows that categories that are specified to be near one another in WordNet, such as the categories related to *bread*, are found to be closely interrelated. This is useful in case we would like to begin with smaller categories, in order to eliminate some of the large, broad categories that we are currently working with.

Most of the connectivity information suggested by the network was used to create the new categories. However, many of the desireable relationships do not appear in the network, perhaps because of the requirement for highly mutual co-ranking. If we were to relax this assumption we may find better coverage, but perhaps at the cost of more misleading links. The remaining associations where determined by hand, so that the original 726 categories were combined into 106 new super-categories. Unlike for the original categories, the names for the super-categories were chosen by the authors.

Revised Topic Assignments

The super-categories are intended to group together related categories in order to eliminate topical redundancy in the labeler and to help eliminate inappropriate labels (since the categories are larger and so have more lexical items serving as evidence). Thus the top four or five super-categories should suffice to indicate the main topics of documents. A comparison of the results of the labeling algorithm using super-categories against reader judgments appears in [9]. Here we show some example output and discuss its characteristics.

Figure 0.5 compares the results of the labeler using the original categories against the super-categories. It should be kept in mind that only the top four or five labels are to be used from the super-categories; since each super-category subsumes many categories, only a few super-categories should be expected to contain the most relevant information. The first article is a 31-sentence magazine article, published in 1987,

taken from [15]. It describes how Soviet women have little political power, discusses their role as working women, and describes the benefits of college life. The second article is a 77-sentence popular science magazine article about the Magellan space probe exploring Venus. When using the super-categories, the labeler avoids grossly inappropriate labels such as "mollusk_genus" and "goddess" in the Magellan article, and combines categories such as "layer", "natural_depression", and "rock stone" into the one super-category "land terra_firma".

Looking again at the longer texts of the *United States Constitution* and *Genesis* we see in Figure 0.6 that the super-categories are more general and less redundant than the categories shown in Figure 0.3. (Although the high scores for the "breads" category seems incorrect, even though the term "bread" occurs 25 times in *Genesis*.) In some cases the user might desire more specific categories; this experiment suggests that the labeler can generate topic labels at multiple levels of granularity.

Conclusions

We have discussed two approaches to augmenting and rearranging the components of a lexicon, in effect adding new features to its members, by making use of lexical association information from a large corpus. We've used lexical cooccurrence statistics in combination with a modified lexicon to classify proper names, associate more specific senses to broadly defined terms, and classify new words into existing categories with some degree of success. We've also used these statistics to suggest how to rearrange a lexicon with a taxonymic structure into more frame-like categories, and assigned more general main-topic labels to texts based on these categories.

One conclusion that may be drawn from this work is that we have provided a mechanism for successfully combining hand-built lexicon information with knowledge-free, statistically-derived information. The combined information from the categories derived from WordNet provided the clusters from which WordSpace centroids could be created, and these centroids in turn provided candidate words to improve the categories.

In future, in addition to expanding the evaluation of the results described here, we would like to try reversing the experiment; that is,

starting with WordSpace vectors, see which parts of WordNet should be interlinked into schematic categories.

Acknowledgments The authors would like to thank Jan Pedersen for his help and encouragement. We are also indebted to Mike Berry for SVDPACK. The first author's research was sponsored in part by the Advanced Research Projects Agency under Grant No. MDA972-92-J-1029 with the Corporation for National Research Initiatives (CNRI), in part by an internship at Xerox Palo Alto Research Center; and this material is based in part upon work supported by the National Science Foundation under Infrastructure Grant No. CDA-8722788. The second author was supported in part by the National Center for Supercomputing Applications under grant BNS930000N.

Bibliography

[1] Hiyan Alshawi. Processing dictionary definitions with phrasal pattern hierarchies. *American Journal of Computational Linguistics*, 13(3):195–202, 1987.

[2] Elan Amir. Carta: A network topology presentation tool. Project Report, UC Berkeley, 1993.

[3] Michael W. Berry. Large-scale sparse singular value computations. *The International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[4] Nicoletta Calzolari and Remo Bindi. Acquisition of lexical information from a large textual italian corpus. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki, 1990.

[5] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[6] T. Fruchtermann and E. Rheingold. Graph drawing by force-directed placement. Technical Report UIUCDCS-R-90-1609, Department of Computer Science, University of Illinois, Urbana-Champagne, Ill, June 1990.

[7] G. Grefenstette. A new knowledge-poor technique for knowledge extraction from large corpora. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, Copenhagen, Denmark, June 21-24 1992. ACM.

[8] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 539–545, Nantes, France, July 1992.

[9] Marti A. Hearst. *Context and Structure in Automated Full-Text Information Access*. PhD thesis, University of California at Berkeley, 1994. (Computer Science Division Technical Report).

[10] Paul Jacobs and Lisa Rau. SCISOR: Extracting information from On-Line News. *Communications of the ACM*, 33(11):88–97, 1990.

[11] Judith Markowitz, Thomas Ahlswede, and Martha Evens. Semantically significant patterns in dictionary definitions. *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 112–119, 1986.

[12] Brij Masand, Gordon Linoff, and David Waltz. Classifying news stories using memory based reasoning. In *Proceedings of ACM/SIGIR*, pages 59–65, 1992.

[13] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.

[14] Marvin Minsky. A framework for representing knowledge. In Patrick Winston, editor, *The psychology of computer vision*. McGraw-Hill, 1975.

[15] Jane Morris. Lexical cohesion, the thesaurus, and the structure of text. Technical Report CSRI-219, Computer Systems Research Institute, University of Toronto, 1988.

[16] James Pustejovsky. On the acquisition of lexical entries: The perceptual origin of thematic relations. *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, 1987.

[17] Philip Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, University of Pennsylvania, December 1993. (Institute for Research in Cognitive Science report IRCS-93-42).

[18] Hinrich Schütze. Word space. In Stephen J. Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, San Mateo CA, 1993.

[19] Yorick A. Wilks, Dan C. Fass, Cheng ming Guo, James E. McDonald, Tony Plate, and Brian M. Slator. Providing machine tractable dictionary tools. *Journal of Computers and Translation*, 2, 1990.

[20] David Yarowsky. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 454–460, Nantes, France, July 1992.

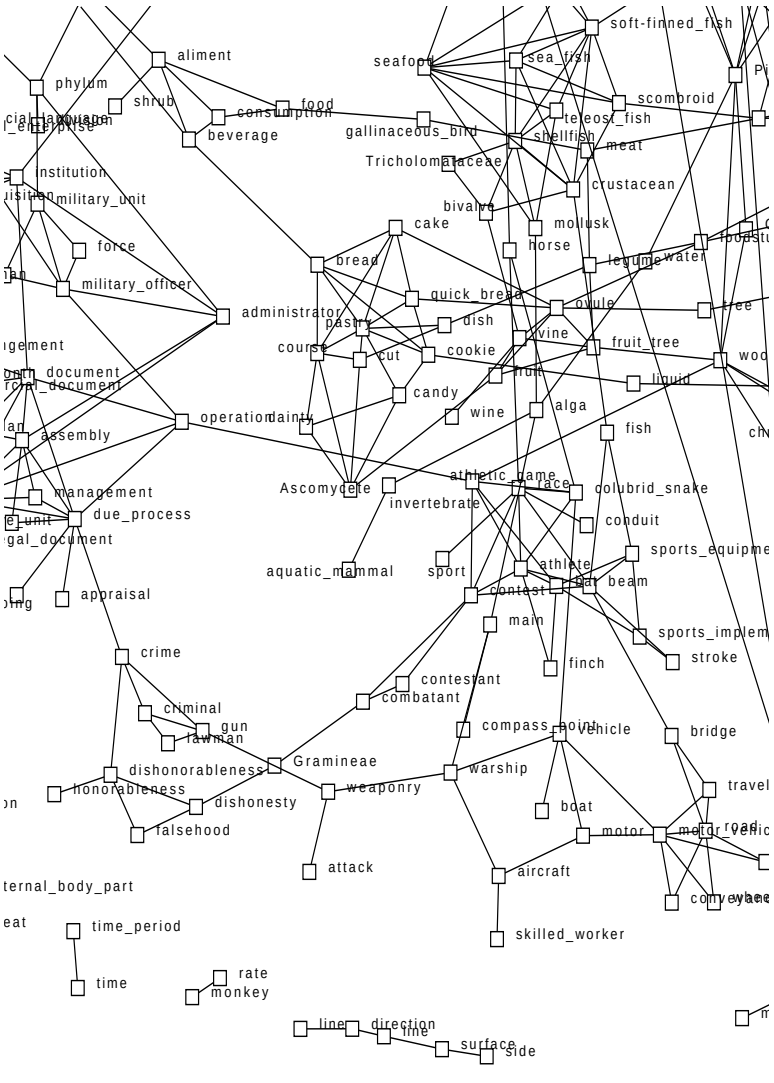

Figure 0.4
A piece of the network that results from determining which categories are mutually close to each other using WordSpace.

Raisa Gorbachev article		
	Original Categories	*Super-Categories*
0	woman adult_female	social_standing
1	status social_state	education
2	man adult_male	politics
3	political_orientation ideology	legal_system
4	force personnel	people
5	charge	psychological_state
6	relationship	socializing
7	fear	social_group
8	attitude	personal_relationship
9	educator pedagogue	government

Magellan space probe article		
	Original Categories	*Super-Categories*
0	celestial_body heavenly_body	outer_space
1	mollusk_genus	light_and_energy
2	electromagnetic_radiation	atmosphere
3	layer (surface)	land terra_firma
4	atmospheric_phenomenon	physics
5	physical_phenomenon	arrangement
6	goddess	shapes
7	natural_depression depression	water_and_liquids
8	rock stone	properties
9	space (hole)	amounts

Figure 0.5
Comparison of original and super categories.

United States Constitution		
	Original Categories	*Super-Categories*
0	assembly (court, legislature)	legal_system
1	due_process_of_law	government
2	legal_document legal_instrument	politics
3	administrative_unit	conflict
4	body (legislative)	crime
5	charge (taxes)	finance
6	administrator decision_maker	social_standing
7	document written_document	honesty
8	approval (sanction, pass)	communication

Genesis		
	Original Categories	*Super-Categories*
0	deity divinity god	religion
1	relative relation (mother, aunt)	breads
2	worship	mythology
3	man adult_male	people
4	professional	social_outcasts
5	happiness gladness felicity	social_group
6	woman adult_female	psychological_state
7	evildoing transgression	personality
8	literary_composition	literature

Figure 0.6
Comparison of original and super categories for two well-known texts.