PART-OF-SPEECH INDUCTION FROM SCRATCH

Hinrich Schütze
Center for the Study of Language and Information
Ventura Hall
Stanford, CA 94305-4115
schuetze@csli.stanford.edu

Abstract

This paper presents a method for inducing the parts of speech of a language and part-of-speech labels for individual words from a large text corpus. Vector representations for the part-of-speech of a word are formed from entries of its near lexical neighbors. A dimensionality reduction creates a space representing the syntactic categories of unambiguous words. A neural net trained on these spatial representations classifies individual contexts of occurrence of ambiguous words. The method classifies both ambiguous and unambiguous words correctly with high accuracy.

INTRODUCTION

Part-of-speech information about individual words is necessary for any kind of syntactic and higher level processing of natural language. While it is easy to obtain lists with part of speech labels for frequent English words, such information is not available for less common languages. Even for English, a categorization of words that is tailored to a particular genre may be desired. Finally, there are rare words that need to be categorized even if frequent words are covered by an available electronic dictionary.

This paper presents a method for inducing the parts of speech of a language and part-of-speech labels for individual words from a large text corpus. Little, if any, language-specific knowledge is used, so that it is applicable to any language in principle. Since the part-of-speech representations are derived from the corpus, the resulting categorization is highly text specific and doesn't contain categories that are inappropriate for the genre in question. The method is efficient enough for vocabularies of tens of thousands of words thus addressing the problem of coverage.

The problem of how syntactic categories can be induced is also of theoretical interest in language acquisition and learnability. Syntactic category information is part of the basic knowledge about language that children must learn before they can acquire more complicated structures. It has been claimed that "the properties that the child can detect in the input – such as the serial positions and adjacency and co-occurrence relations among words – are in general linguistically irrelevant." (Pinker 1984) It will be shown here that relative position of words with respect to each other is sufficient for learning the major syntactic categories.

In the first part of the derivation, two iterations of a massive linear approximation of cooccurrence counts categorize unambiguous words. Then a neural net trained on these words classifies individual contexts of occurrence of ambiguous words. An evaluation suggests that the method classifies both ambiguous and unambiguous words correctly. It differs from previous work in its efficiency and applicability to large vocabularies; and in that linguistic knowledge is only used in the very last step so that theoretical assumptions that don't hold for a language or sublanguage have minimal influence on the classification.

The next two sections describe the linear approximation and a *birecurrent* neural network for the classification of ambiguous words. The last section discusses the results.

CATEGORY SPACE

The goal of the first step of the induction is to compute a multidimensional real-valued space, called *category space*, in which the syntactic category of each word is represented by a vector. Proximity in the space is related to similarity of syntactic category. The vectors in this space will then be used as input and target vectors for the connectionist net.

The vector space is bootstrapped by collecting relevant distributional information about words. The 5,000 most frequent words in five months of the New York Times News Service (June through

October 1990) were selected for the experiments. For each pair of these words $< w_i, w_j >$, the number of occurrences of w_i immediately to the left of w_j ($b_{i,j}$), the number of occurrences of w_i immediately to the right of w_j ($c_{i,j}$), the number of occurrences of w_i at a distance of one word to the left of w_j ($a_{i,j}$), and the number of occurrences of w_i at a distance of one word to the right of w_j ($d_{i,j}$) were counted. The four sets of 25,000,000 counts were collected in the 5,000-by-5,000 matrices B, C, A, and D, respectively. Finally these four matrices were combined into one large 5,000-by-20,000 matrix as shown in Figure 1. The figure also shows for two words where their four cooccurrence counts are located in the 5,000-by-20,000 matrix. In the experiments, w_{3000} was *resistance* and w_{4250} was *theaters*. The four marks in the figure, the positions of the counts $a_{3000,4250}$, $b_{3000,4250}$, $c_{3000,4250}$, and $d_{3000,4250}$, indicate how often *resistance* occurred at positions -2, -1, 1, and 2 with respect to *theaters*.

These 20,000-element rows of the matrix could be used directly to compute the syntactic similarity between individual words: The cosine of the angle between the vectors of a pair of words is a measure of their similarity.[1] However, computations with such large vectors are time-consuming. Therefore a singular value decomposition was performed on the matrix. Fifteen singular values were computed using a sparse matrix algorithm from SVDPACK (Berry 1992). As a result, each of the 5,000 words is represented by a vector of real numbers. Since the original 20,000-component vectors of two words (corresponding to rows in the matrix in Figure 1) are similar if their collocations are similar, the same holds for the reduced vectors because the singular value decomposition finds the best least square approximation for the 5,000 original vectors in a 15-dimensional space that preserves similarity between vectors. See (Deerwester et al. 1990) for a definition of SVD and an application to a similar problem.

Close neighbors in the 15-dimensional space generally have the same syntactic category as can be seen in Table 1. However, the problem with this method is that it will not scale up to a very large number of words. The singular value decomposition has a time complexity quadratic in the rank of the matrix, so that one can only treat a small part of the total vocabulary of a large corpus.

Therefore, an alternative set of features was considered: *classes* of words in the 15-dimensional space. Instead of counting the number of occurrences of individual words, we would now count the number of occurrences of members of word classes.[2] The space was clustered with Buckshot, a linear-time clustering algorithm described in (Cutting et al. 1992). Buckshort applies a high-quality quadratic clustering algorithm to a random sample of size $\sqrt{kn}$, where k is the number of desired cluster centers and n is the number of vectors to be clustered. Each of the remaining $n - \sqrt{kn}$ vectors is assigned to the nearest cluster center. The high-quality quadratic clustering algorithm used was truncated group average agglomeration (Cutting et al. 1992).

Clustering algorithms generally do not construct groups with just one member. But there are many closed-class words such as auxiliaries and prepositions that shouldn't be thrown together with the open classes (verbs, nouns etc.). Therefore, a list of 278 closed-class words, essentially the words with the highest frequency, was set aside. The remaining 4722 words were classified into 222 classes using Buckshot.

The resulting 500 classes (278 high-frequency words, 222 clusters) were used as features in the matrix shown in Figure 2. Since the number of features has been greatly reduced, a larger number of words can be considered. For the second matrix all 22,771 words that occurred at least 100 times in 18 months of the New York Times News Service (May 1989 – October 1990) were selected. Again, there are four submatrices, corresponding to four relative positions. For example, the entries $a_{i,j}$ in the A part of the matrix count how often a member of class i occurs at a distance of one word to the left of word j. Again, a singular value decomposition was performed on the matrix, this time 10 singular values were computed. (Note that in the first figure the 20,000-element *rows* of the matrix are reduced to 15 dimensions whereas in the second matrix the 2,000-element *columns* are reduced to 10 dimensions.)

Table 2 shows 20 randomly selected words and their nearest neighbors in category space (in order of proximity to the head word). As can be seen from the table, proximity in the space is a good predictor of similar syntactic category. The nearest neighbors of *athlete*, *clerk*, *declaration*, and *dome* are singular nouns, the nearest neighbors of *bowers* and *gibbs* are family names, the nearest neighbors of *desirable* and *sole* are adjectives, and the nearest neighbors of *financings* are plural nouns, in each case without exception. The neighborhoods of *armaments*, *cliches* and *luxuries* (nouns), and *b'nai* and *northwestern* (NP-initial modifiers) fail to respect finer grained syntactic

[1] The cosine between two vectors corresponds to the normalized correlation coefficient: $\cos(\alpha(\vec{x}, \vec{y})) = \sum_i (x_i y_i) / \sqrt{\sum_i x_i^2 \sum_i y_i^2}$.

[2] Cf. (Brown et al. 1992) where the same idea of improving generalization and accuracy by looking at word classes instead of individual words is used.

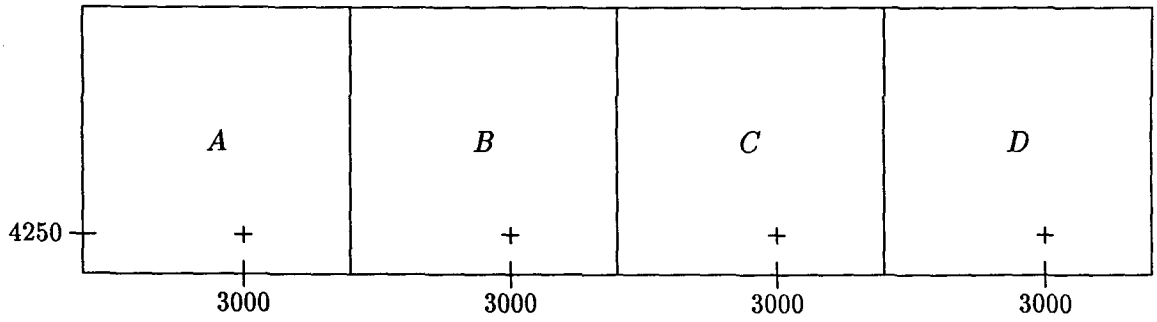

Figure 1: The setup of the matrix for the first singular value decomposition.

Table 1: Ten random and three selected words and their nearest neighbors in category space 1.

word	nearest neighbors
accompanied	submitted banned financed developed authorized headed canceled awarded barred
almost	virtually merely formally fully quite officially just nearly only less
causing	reflecting forcing providing creating producing becoming carrying particularly
classes	elections courses payments losses computers performances violations levels pictures
directors	professionals investigations materials competitors agreements papers transactions
goal	mood roof eye image tool song pool scene gap voice
japanese	chinese iraqi american western arab foreign european federal soviet indian
represent	reveal attend deliver reflect choose contain impose manage establish retain
think	believe wish know realize wonder assume feel say mean bet
york	angeles francisco sox rouge kong diego zone vegas inning layer
on	through in at over into with from for by across
must	might would could cannot will should can may does helps
they	we you i he she nobody who it everybody there

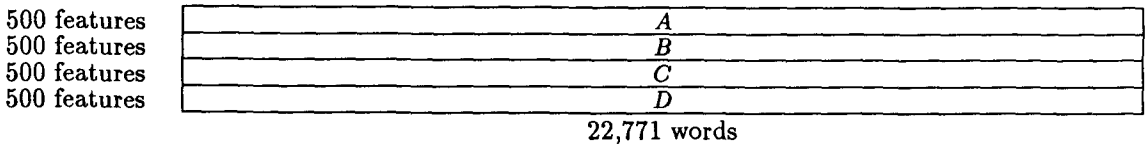

Figure 2: The setup of the matrix for the second singular value decomposition.

Table 2: Twenty random and four selected words and their neigborhoods in category space 2.

word	nearest neighbors
armaments	turmoil weaponry landmarks coordination prejudices secrecy brutality unrest harassment |
athlete	virus scenario | event audience disorder organism candidate procedure epidemic
b'nai	| suffolk sri allegheny cosmopolitan berkshire cuny broward multimedia bovine nytimes
bowers	jacobs levine carr hahn schwartz adams buckley dershowitz fitzpatrick peterson |
clerk	salesman | psychologist photographer preacher mechanic dancer lawyer trooper trainer
cliches	pests wrinkles outbursts streams icons endorsements | friction unease appraisals lifestyles
cruz	antonio | clara pont saud monica paulo rosa mae attorney palma
declaration	sequence mood profession marketplace concept facade populace downturn moratorium |
desirable	recognizable | frightening loyal devastating exciting troublesome awkward palpable
dome	blackout furnace temblor quartet citation chain countdown thermometer shaft |
equally	| somewhat progressively acutely enormously excessively unnecessarily largely scattered
financings	| endeavors monopolies raids patrols stalls offerings occupations philosophies religions
gibbs	adler reid webb jenkins stevens carr laurent dempsey hayes farrell |
luxuries	volatility insight hostility dissatisfaction stereotypes competence unease animosity residues |
northwestern	| baja rancho harvard westchester ubs humboldt laguna guinness vero granada
oh	gee gosh ah hey | appleton ashton dolly boldface baskin lo
sole	| lengthy vast monumental rudimentary nonviolent extramarital lingering meager gruesome
transports	| spokesman copyboy staffer barrios comptroller alloy stalks spokeswoman dal spokesperson
vividly	|skillfully frantically calmly confidently streaming relentlessly discreetly spontaneously
walks	floats | jumps collapsed sticks stares crumbled peaked disapproved runs crashed
claims	credits promises | forecasts shifts searches trades practices processes supplements controls
on	through from in | at by within with under against for
must	will might would cannot could can should won't | doesn't may
they	we | i you who nobody he it she everybody there

distinctions, but are reasonable representations of syntactic category. The neighbors of *cruz* (second components of names), and *equally* and *vividly* (adverbs) include words of the wrong category, but are correct for the most part.

In order to give a rough idea of the density of the space in different locations, the symbol "|" is placed before the first neighbor in Table 2 that has a correlation of 0.978 or less with the head word. As can be seen from the table, the regions occupied by nouns and proper names are dense, whereas adverbs and adjectives have more distant nearest neighbors. One could attempt to find a fixed threshold that would separate neighbors of the same category from syntactically different ones. For instance, the neighbors of *oh* with a correlation higher than 0.978 are all interjections and the neighbors of *cliches* within the threshold region are all plural nouns. However, since the density in the space is different for different regions, it is unlikely that a general threshold for all syntactic categories can be found.

The neighborhoods of *transports* and *walks* are not very homogeneous. These two words are ambiguous between third person singular present tense and plural noun. Ambiguity is a problem for the vector representation scheme used here, because the two components of an ambiguous vector can add up in a way that makes it by chance similar to an unambiguous word of a different syntactic category. If we call the distributional vector $\vec{v}_c$ of words of category c the *profile* of category c, and if a word w_1 is used with frequency α in category c_1 and with frequency β in category c_2, then the weighted sum of the profiles (which corresponds to a column for word w_1 in Figure 2) may turn out to be the same as the profile of an unrelated third category c_3:

$$\alpha\vec{v}_{c_1} + \beta\vec{v}_{c_2} = \vec{v}_{c_3}$$

This is probably what happened in the cases of *transports* and *walks*. The neighbors of *claims* demonstrate that there are homogeneous "ambiguous" regions in the space if there are enough words with the same ambiguity and the same frequency ratio of the categories. *transports* and *walks* (together with *floats*, *jumps*, *sticks*, *stares*, and *runs*) seem to have frequency ratios α/β different from *claims*, so that they ended up in different regions.

The last three lines of Table 2 indicate that function words such as prepositions, auxiliaries, and nominative pronouns and quantifiers occupy their own regions, and are well separated from each other and from open classes.

A BIRECURRENT NETWORK FOR PART-OF-SPEECH PREDICTION

A straightforward way to take advantage of the vector representations for part of speech categorization is to cluster the space and to assign part-of-speech labels to the clusters. This was done with Buckshot. The resulting 200 clusters yielded good results for unambiguous words. However, for the reasons discussed above (linear combination of profiles of different categories) the clustering was not very successful for ambiguous words. Therefore, a different strategy was chosen for assigning category labels. In order to tease apart the different uses of ambiguous words, one has to go back to the individual contexts of use. The connectionist network in Figure 3 was used to analyze individual contexts.

The idea of the network is similar to Elman's recurrent networks (Elman 1990, Elman 1991): The network learns about the syntactic structure of the language by trying to predict the next word from its own context units in the previous step and the current word. The network in Figure 3 has two novel features: It uses the vectors from the second singular value decomposition as input and target. Note that distributed vector representations are ideal for connectionist nets, so that a connectionist model seems most appropriate for the prediction task. The second innovation is that the net is *birecurrent*. It has recurrency to the left as well as to the right.

In more detail, the network's input consists of the word to the left t_{n-1}, its own left context in the previous time step $c\text{-}l_{n-1}$, the word to the right t_{n+1} and its own right context $c\text{-}r_{n+1}$ in the next time step. The second layer has the context units of the current time step. These feed into thirty hidden units h_n which in turn produce the output vector o_n. The target is the current word t_n. The output units are linear, hidden units are sigmoidal.

The network was trained stochastically with truncated backpropagation through time (BPTT, Rumelhart et al. 1986, Williams and Peng 1990). For this purpose, the left context units were unfolded four time steps to the left and the right context units four time steps to the right as shown in Figure 4. The four blocks of weights on the connections to $c\text{-}l_{n-3}$, $c\text{-}l_{n-2}$, $c\text{-}l_{n-1}$, and $c\text{-}l_n$ are linked to ensure identical mapping from one "time step" to the next. The connections on the right side are linked in the same way. The training set consisted of 8,000 words in the New York Times newswire (from June 1990). For each training step, four words to the left of the target word ($t_{n-3}, t_{n-2}, t_{n-1}$, and t_n) and four words to the right of the target word (t_n, t_{n+1}, t_{n+2}, and t_{n+3})

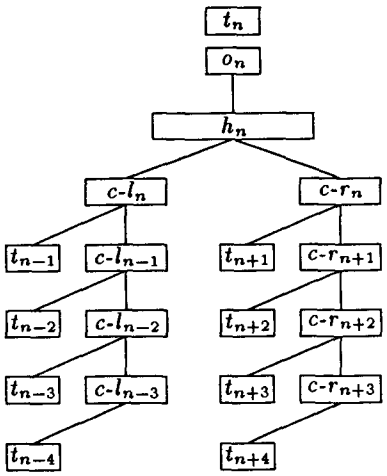

Figure 4: Unfolded birecurrent network in training.

were the input to the unfolded network. The target was the word t_n. A modification of bp from the pdp package was used with a learning rate of 0.01 for recurrent units, 0.001 for other units and no momentum.

After training, the network was applied to the category prediction tasks described below by choosing a part of the text without unknown words, computing all left contexts from left to right, computing all right contexts from right to left, and finally predicting the desired category of a word t_n by using the precomputed contexts $c\text{-}l_n$ and $c\text{-}r_n$.

In order to tag the occurrence of a word, one could retrieve the word in category space whose vector is closest to the output vector computed by the network. However, this would give rise to too much variety in category labels. To illustrate, consider the prediction of the category NOUN. If the network categorizes occurrences of nouns correctly as being in the region around *declaration*, then the slightest variation in the output will change the nearest neighbor of the output vector from *declaration* to its nearest neighbors *sequence* or *mood* (see Table 2). This would be confusing to the human user of the categorization program.

Therefore, the first 5,000 output vectors of the network (from the first day of June 1990), were clustered into 200 *output clusters* with Buckshot. Each output cluster was labeled by the two words closest to its centroid. Table 3 lists labels of some of the output clusters that occurred in the experiment described below. They are easily interpretable for someone with minimal linguistic knowledge as the examples show. For some categories such as HIS_THE one needs to look at a couple of instances to get a "feel" for their mean-

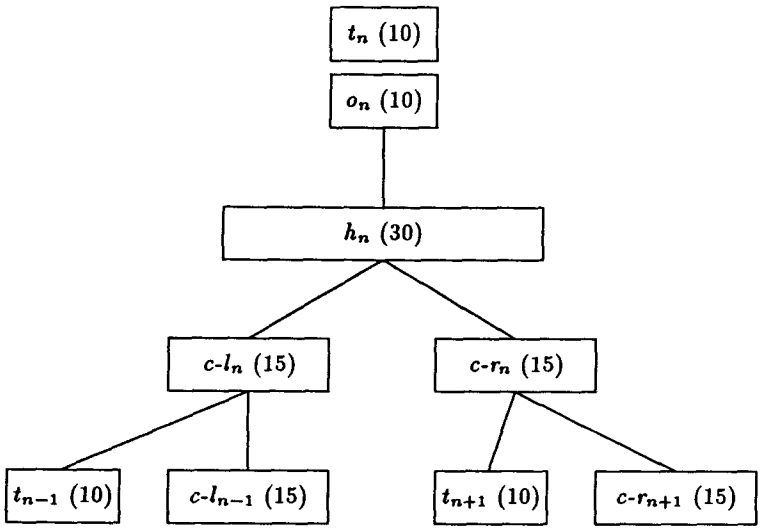

Figure 3: The architecture of the birecurrent network

Table 3: The labels of 10 output clusters.

output cluster label	part of speech
excel_depart	intransitive verb (base form)
prompt_select	transitive verb (base form)
cares_sounds	3. person sg. present tense
office_staff	noun
promotion_trauma	noun
famous_talented	adjective
publicly_badly	adverb
his_the	NP-initial

ing.

The syntactic distribution of an individual word can now be more accurately determined by the following algorithm:

- compute an output vector for each position in the text at which the target word occurs.
- for each output vector j do the following:
  - determine the centroid of the cluster i which is closest
  - compute the correlation coefficient of the output vector j and the centroid of the output cluster i. This is the score $s_{i,j}$ for cluster i and vector j. Assign zero to the scores of the other clusters for this vector: $s_{k,j} := 0, k \neq i$
- for each cluster i, compute the final score f_i as the sum of the scores $s_{i,j}$: $f_i := \sum_j s_{i,j}$
- normalize the vector of 200 final scores to unit length

This algorithm was applied to June 1990. If for a given word, the sum of the unnormalized final scores was less than 30 (corresponding to roughly 100 occurrences in June), then this word was discarded. Table 4 lists the highest scoring categories for 10 random words and 11 selected ambiguous words. (Only categories with a score of at least 0.2 are listed.)

The network failed to learn the distinctions between adjectives, intransitive present participles and past participles in the frame "*to-be* + □ + non-NP". For this reason, the adjective *close*, the present participle *beginning*, and the past participle *shot* are all classified as belonging to the category STRUGGLING_TRAVELING. (Present Participles are successfully discriminated in the frame "*to-be* + □ + NP": see *winning* in the table, which is classified as the progressive form of a transitive verb: HOLDING_PROMISING.) This is the place where linguistic knowledge has to be injected in form of the following two rules:

- If a word in STRUGGLING_TRAVELING is a morphological present participle or past participle assign it to that category, otherwise to the category ADJECTIVE_PREDICATIVE.
- If a word in a noun category is a morphological plural assign it to NOUN_PLURAL, to NOUN_SINGULAR otherwise.

With these two rules, all major categories are among the first found by the algorithm; in particular the major categories of the ambiguous words *better* (adjective/adverb), *close* (verb/adjective), *work* (noun/base form of verb), *hopes* (noun/third person singular), *beginning* (noun/present-participle), *shot* (noun/past participle) and *'s* ('s/is). There are two clear errors: GIVEN_TAKING for *contain*, and RICAN_ADVISORY for *'s*, both of rank three in the table.

Table 4: The highest scoring categories for 10 random and 11 selected words.

word	highest scoring categories		
adequate	universal_martial (0.50)	struggling_traveling (0.33)	several_numerous (0.33)
admit	excel_depart (0.88)	gather_propose (0.30)	prompt_select (0.20)
appoint	prompt_select (0.72)	gather_propose (0.65)	
consensus	office_staff (0.71)	promotion_trauma (0.43)	hand_shooting (0.39)
contain	gather_propose (0.76)	prompt_select (0.43)	given_taking (0.24)
dodgers	promotion_trauma (0.57)	yankees_paper (0.52)	fantasy_ticket (0.48)
genes	office_staff (0.43)	promotion_trauma (0.75)	route_style (0.22)
language	promotion_trauma (0.65)	office_staff (0.57)	office_agent (0.21)
legacy	promotion_trauma (0.95)	office_staff (0.22)	
thirds	hand_shooting (0.75)	famous_talented (0.41)	iron_pickup (0.36)
good	famous_talented (0.86)		
better	famous_talented (0.65)	his_the (0.34)	publicly_badly (0.27)
close	gather_propose (0.43)	struggling_traveling (0.42)	famous_talented (0.36)
work	excel_depart (0.72)	promotion_trauma (0.51)	remain_want (0.27)
hospital	promotion_trauma (0.75)	office_agent (0.40)	fantasy_ticket (0.24)
buy	gather_propose (0.77)	prompt_select (0.47)	remain_want (0.22)
hopes	promotion_trauma (0.56)	cares_sounds (0.53)	windows_pictures (0.21)
beginning	promotion_trauma (0.90)	struggling_traveling (0.34)	
shot	hand_shooting (0.54)	struggling_traveling (0.45)	promotion_trauma (0.40)
's	's_facto (0.54)	makes_is (0.40)	rican_advisory (0.37)
winning	famous_talented (0.71)	holding_promising (0.33)	iron_pickup (0.29)

These results seem promising given the fact that the context vectors consist of only 15 units. It seems naive to believe that all syntactic information of the sequence of words to the left (or to the right) can be expressed in such a small number of units. A larger experiment with more hidden units for each context vector will hopefully yield better results.

DISCUSSION AND CONCLUSION

Brill and Marcus describe an approach with similar goals in (Brill and Marcus 1992). Their method requires an initial consultation of a native speaker for a couple of hours. The method presented here makes a short consultation of a native speaker necessary, however it occurs at the end, as the last step of category induction. This has the advantage of avoiding bias in an initial a priori classification.

Finch and Chater present an approach to category induction that also starts out with offset counts, proceeds by classifying words on the basis of these counts, and then goes back to the local context for better results (Finch and Chater 1992). But the mathematical and computational techniques used here seem to be more efficient and more accurate than Finch and Chater's, and hence applicable to vocabularies of a more realistic size.

An important feature of the last step of the procedure, the neural network, is that the lexicographer or linguist can browse the space of output vectors for a given word to get a sense of its syntactic distribution (for instance uses of *better* as an adverb) or to improve the classification (for instance by splitting an induced category that is too coarse). The algorithm can also be used for categorizing unseen words. This is possible as long as the words surrounding it are known.

The procedure for part-of-speech categorization introduced here may be of interest even for words whose part-of-speech labels are known. The dimensionality reduction makes the global distributional pattern of a word available in a profile consisting of a dozen or so real numbers. Because of its compactness, this profile can be used efficiently as an additional source of information for improving the performance of natural language processing systems. For example, adverbs may be lumped into one category in the lexicon of a processing system. But the category vectors of adverbs that are used in different positions such as *completely* (mainly pre-adjectival), *normally* (mainly pre-verbal) and *differently* (mainly post-verbal) are different because of their different distributional properties. This information can be exploited by a parser if the category vectors are available as an additional source of information.

The model has also implications for language acquisition. (Maratsos and Chalkley 1981) propose that the *absolute* position of words in sentences is important evidence in children's learning of categories. The results presented here show that *relative* position is sufficient for learning the major syntactic categories. This suggests that relative position could be important information for learning syntactic categories in child language acquisition.

The basic idea of this paper is to collect a

large amount of distributional information consisting of word cooccurrence counts and to compute a compact, low-rank approximation. The same approach was applied in (Schütze, forthcoming) to the induction of vector representations for semantic information about words (a different source of distributional information was used there). Because of the graded information present in a multi-dimensional space, vector representations are particularly well-suited for integrating different sources of information for disambiguation.

In summary, the algorithm introduced here provides a language-independent, largely automatic method for inducing highly text-specific syntactic categories for a large vocabulary. It is to be hoped that the method for distributional analysis presented here will make it easier for computational and traditional lexicographers to build dictionaries that accurately reflect language use.

ACKNOWLEDGMENTS

I'm indebted to Mike Berry for SVDPACK and to Marti Hearst, Jan Pedersen and two anonymous reviewers for very helpful comments. This work was partially supported by the National Center for Supercomputing Applications under grant BNS930000N.

REFERENCES

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

Brill, Eric, and Mitch Marcus. 1992. Tagging an Unfamiliar Text with Minimal Human Supervision. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, ed. Robert Goldman. AAAI Press.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-Based *n*-gram Models of Natural Language. *Computational Linguistics* 18(4):467–479.

Cutting, Douglas R., Jan O. Pedersen, David Karger, and John W. Tukey. 1992. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of SIGIR '92.*

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

Elman, Jeffrey L. 1990. Finding Structure in Time. *Cognitive Science* 14:179–211.

Elman, Jeffrey L. 1991. Distributed Representations, Simple Recurrent Networks, and Grammatical Structure. *Machine Learning* 7(2/3):195–225.

Finch, Steven, and Nick Chater. 1992. Bootstrapping Syntactic Categories Using Statistical Methods. In *Background and Experiments in Machine Learning of Natural Language*, ed. Walter Daelemans and David Powers. Tilburg University. Institute for Language Technology and AI.

Maratsos, M. P., and M. Chalkley. 1981. The internal language of children's syntax: the ontogenesis and representation of syntactic categories. In *Children's language*, ed. K. Nelson. New York: Gardner Press.

Pinker, Steven. 1984. *Language Learnability and Language Development.* Cambridge MA: Harvard University Press.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Volume 1: Foundations*, ed. David E. Rumelhart, James L. McClelland, and the PDP Research Group. Cambridge MA: The MIT Press.

Schütze, Hinrich. Forthcoming. Word Space. In *Advances in Neural Information Processing Systems 5*, ed. Stephen J. Hanson, Jack D. Cowan, and C. Lee Giles. San Mateo CA: Morgan Kaufmann.

Williams, Ronald J., and Jing Peng. 1990. An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories. *Neural Computation* 2:490–501.