Word Sense Disambiguation With Sublexical Representations

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

Abstract

This paper introduces a new representational scheme for word sense disambiguation. Drawing on work in information retrieval (Latent Semantic Indexing as proposed by [Deerwester *et al.* 1990]) an efficient method for learning *sublexical representations* is described: Words and contexts are represented as vectors in a multidimensional space which approximates similarity of collocational patterns. Closeness of words in the space is equivalent to occurrence in similar contexts, thus giving a rough approximation of semantic similarity.

The Bayesian classification system AutoClass was then used to perform an unsupervised classification of sublexical representations of contexts of the three ambiguous words *interest*, *suit* and *plant* in a training text. In applying this classification to a test text, AutoClass disambiguated 90% of all occurrences correctly. Unsupervised classification failed for *tank*, but a more sophisticated algorithm also achieved a disambiguation rate of 90%.

Introduction

In his 1988 paper *Distributed representations of ambiguous words and their resolution in a connectionist network*, Alan Kawamoto shows that vector representations of words can account for many findings in lexical access and lexical priming that have been reported in the psycholinguistic literature [Kawamoto 1988]. The crucial property of the representations he uses is that words with similar spellings, pronunciations, syntactic forms and meanings are represented by similar vectors, i.e. by vectors with a high correlation in their components or, on a geometric interpretation, vectors forming a small angle in a multidimensional space.

If we want to use vectors for word sense disambiguation, the first step is to find a representational scheme that respects this similarity constraint for lexical meaning. (We will only deal with semantic information here.) This problem is comparable to defining a good measure of similarity between documents in information retrieval. One of the standard solutions is to represent a document by a vector of term counts: The ith component of a document vector contains the number of occurrences of term i in the document [Salton,McGill 1983]. An approximation of how close in content two documents are is then how many components of their respective vectors are similar, i.e. how many terms are used with a similar frequency.

This idea can be applied to approximating the semantics of a word by counting the number of occurrences of a set of terms in windows of a given size around this word in a text corpus. An example is the column headed by *bank* in the (fictitious) collocation matrix shown in Figure 1. *soar* and *sport* are terms and *bank*, *interest* and *beat* are words. Each entry in the column *bank* is a cooccurrence count: The two entries shown encode the information that *soar* occurs 300 times in windows around *bank* and that *sport* occurs 75 times in windows around *bank*. According to the correlation measure, *bank* and *interest* and *beat* and *interest* are similar since they have similar counts, but *bank* and *beat* are less similar since their counts don't correlate very well. Formally, the correlation coefficient can be computed as follows: [Salton,McGill 1983]

$$\mathrm{COS}(\mathrm{WORD}_i, \mathrm{WORD}_j) = \frac{\sum_{k=1}^{n}(a_{k,i}a_{k,j})}{\sqrt{\sum_{k=1}^{n} a_{k,i}^2 \sum_{k=1}^{n} a_{k,j}^2}}$$

This gives the following results for the three word pairs in Figure 1: COS(*bank* , *interest*) = 0.94, COS(*interest*,*beat*)=0.92, COS(*bank*,*beat*)=0.74.

The collocation matrix can be interpreted geometrically as shown in Figure 2. Terms are axes, words are vectors whose components on the various dimensions are determined by the cooccurrence counts in the collocation matrix. Similarity between vectors has then a straightforward visual equivalent: Closeness in the

	BANK	INTEREST	BEAT
SOAR	300	210	133
SPORT	75	140	200

Figure 1: A collocation matrix.

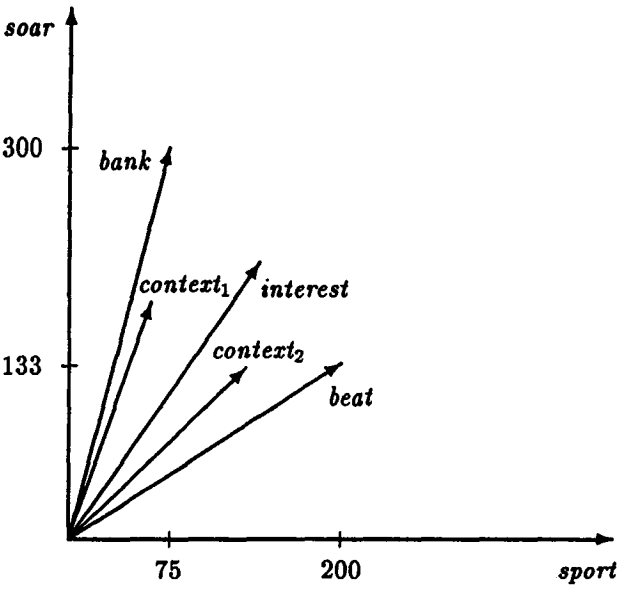

Figure 2: A vector model for context.

multidimensional space corresponding to the collocation matrix. In Figure 2 *bank* and *beat* are not very close to each other, but both are close to the vector *interest* between them.

In order to use the concept of similarity for disambiguation, a vector representation for context is needed that meshes with the representations of words based on collocations. A simple scheme is to compute as the representation for the context at a given position in the text the centroid or average of the vectors of the words close to that position. To see how to disambiguate in this setup consider the example of *interest*. Let us use the tags PERCENT for the sense "charge on borrowed money" and CONCERN for "a feeling that accompanies or causes special attention." Then the PERCENT sense will occur more often in contexts that score high on the *soar* dimension since it is usually interest rates that soar (at least in the corpus used for this study: the New York Times). On the other hand, *sport* will cooccur with the CONCERN sense more often than with the PERCENT sense. We can then disambiguate an occurrence of *interest* at a given position in the text by computing the context vector of that position and comparing how close it is to the *soar* and *sport* dimensions of the space. Two such context vectors are depicted in Figure 2. Vector $context_1$ is closer to *soar*, so probably it is an occurrence of the PERCENT sense of *interest*. Vector $context_2$ is closer to *sport*, and it will be an occurrence of the CONCERN sense.

Of course, it wouldn't work to consider only *soar* and *sport* as terms. More terms are necessary for good results. But the basic idea works fairly well as will be shown below.

For simplicity, I have used the same lexical items both as terms and words in the experiments described in this paper. Every lexical item corresponds then to a dimension in a multidimensional space, in its term role; but also to a vector, in its word role.

Sublexical analysis

There are two problems with the collocational vector representations described above: The data they are based on are noisy; and they take up a lot of space. Similar problems occur with document spaces in information retrieval. In their 1990 paper *Indexing by latent semantic analysis*, Deerwester, Dumais, Furnas, Landauer, and Harshman propose to solve these problems for document spaces by approximating the high dimensional original document space with a lower dimensional space [Deerwester *et al.* 1990]. They use the regression technique of singular value decomposition for this purpose. Applied to our case, it amounts to decomposing the collocation matrix C into three matrices T_0, S_0, and D_0 such that:

$$C = T_0 S_0 D_0'$$

S_0 is a diagonal matrix that contains the singular values of C in descending order. The ith singular value can be interpreted as indicating the strength of the ith principal component of C. T_0 and D_0 are uniquely determined orthonormal matrices that approximate the rows and columns of C, respectively. By restricting the $n \times n$ matrices T_0, S_0, and D_0 to their first $m < n$ columns (= principal components) one obtains the matrices T, S, and D. Their product $\hat{C}$ is the best least square approximation of C in an m-dimensional space:

$$\hat{C} = TSD'$$

We can thus overcome the problems of noisiness and space inefficiency by using the columns of D for the columns of C as vector representations for the selected lexical items. See [Deerwester *et al.* 1990, Golub,Van Loan 1989] for a more detailed description of singular value decomposition and [Schütze Forthcomingb] for an analysis of the induced representations.

Deerwester *et al.* suggest interpreting the process of projecting the original n-space to the reduced m-space as uncovering a *latent semantic structure* of the document space. The corresponding conjecture for the collocation matrix is that the regression gets at an underlying structure of the lexicon that lies beneath the surface of the words and can be uncovered by looking at the set of contexts that words occur in; hence the term *sublexical representations*. Sublexical representations are distributed in the sense of [Rumelhart,McClelland 1986, van Gelder 1991] and can thus be seen as constituting a subsymbolic level from which the symbolic meanings of words may emerge. (cf. [Smolensky 1988, Schütze Forthcominga])

In what follows a disambiguation algorithm based on sublexical representations will be described. The corpus used is the New York Times, June through November 1990 with about four million words per month. The singular value decompositions were computed at the San Diego Supercomputer Center using SSVDC from LINPACK for *interest*, and LAS1 from Michael Berry's SVDPACK for the other three words [Berry 1992].

In the case of *interest*, the following steps were taken:

1. select 977 *training* words that frequently cooccur with *interest*;
2. compute the 977 × 977 collocation matrix for these words by counting the cooccurrences in the training text (July and August 1990) on the basis of windows of size 21;
3. decompose the collocation matrix;
4. extract the first ten principal components;
5. calculate the context vectors for the 1418 occurrences of *interest* in the training text;
6. select the occurrences in the training text that have at least 8 training words in their context.

There are three possibilities for disambiguating new occurrences of *interest* using the set of contexts in 6:

- **Disambiguation by nearest neighbor.** For a new context, find the closest context in the training text and assign its sense to the new context.
- **Supervised classification.** Disambiguate all contexts in the training text, and classify the context set trying to find homogeneous classes. Assign to a new context the sense of the class it falls into.
- **Unsupervised classification.** Classify the raw context vectors in the training text and assign senses to the classes found. Assign to a new context the sense of the class it falls into.

For the first three polysemes, unsupervised classification yielded fairly good results. The classification was done with AutoClass which was provided by NASA Ames Research Center and RIACS [Cheeseman *et al.* 1988]. Assuming a normal distribution for the data, AutoClass found two classes which corresponded closely to the two basic senses PERCENT and CONCERN mentioned above. 135 occurences of *interest* chosen from articles in August 1990 that had not been used for training were then assigned to either of the two classes by AutoClass. The results are shown in Table 1. 2% of the contexts in the training text were either repetitions of previous contexts or represented the rare sense "legal share."

		CONCERN	PERCENT	total
senses in %		77	23	
correct	#	91	28	119
	%	89	93	90
incorrect	#	11	2	13
	%	11	7	10
total	#	102	30	132
	%	100	100	100

Table 1: Disambiguation results for *interest*.

The setup for all four disambiguation experiments is summarized in Table 2. "Unsupervised classification" is abbreviated with "C", the nearest neighbor method with "NN".

AutoClass also found a good classification for *suit* and *plant*. The rare senses for *suit* that were excluded were: "all the playing cards in a pack bearing the same symbol", "the suit led (follow suit)", and "to be proper for." Disambiguation results are listed in Table 3.

		LAWSUIT	GARMENT	total
senses in %		54	46	
correct	#	126	110	236
	%	95	96	96
incorrect	#	7	4	11
	%	5	4	4
total	#	133	114	247
	%	100	100	100

Table 3: Disambiguation results for *suit*.

The disambiguation results for *plant* are shown in Table 4. Metaphorical uses like "to plant a bomb", "to plant a kiss" or "to plant a suction cup Garfield" are not included in the table.

tank turned out to be a difficult case. The classification found by AutoClass was about 85% right for the VEHICLE sense, but more than 50% of RECEPTACLE contexts were misclassified. One reason may be

	interest	*suit*	*plant*	*tank*
# training words	977	2086	3000	4441
training text	Jul–Aug	Jun–Oct	Jun–Oct	Jun–Oct
# occurrences	1328	1332	3382	1343
window size	21	31	31	21
# principal components	10	14	11	13
disambiguation method	C	C	C	NN
# classes	2	3	10	–
test text	Aug	Nov	Nov	Nov
# occurrences	135	290	200	251
% doublets & rare senses	2	15	6	14
% correct	90	96	90	91

Table 2: Four disambiguation experiments.

senses in %		INDUSTRY 64	BOTANY 36	total
correct	#	119	50	169
	%	99	74	90
incorrect	#	1	18	19
	%	1	26	10
total	#	120	68	188
	%	100	100	100

Table 4: Disambiguation results for *plant*.

senses in %		VEHICLE 81	RECEPTACLE 19	total
correct	#	≈ 168	29	197
	%	95	73	91
incorrect	#	≈ 8	11	19
	%	5	28	9
total	#	176	40	216
	%	100	100	100

Table 5: Disambiguation results for *tank*.

that the 4441 training words were selected according to frequency of occurrence with *tank* or *interest*. Since *interest* is much more frequent, most of the 4441 words are not relevant for *tank*. But the main problem with *tank* is that its subordinate sense RECEPTACLE is used for three quite different types of objects: water treatment tanks, gasoline tanks, and tanks for breeding fish. It seems likely that these senses are expressed by different words in other languages. Only about 250 RECEPTACLE contexts occurred in the training text. Words typical for the subordinate sense of *tank* therefore only had a small chance of being included in the 4441 training words. As a result, the semantic fields corresponding to the three RECEPTACLE senses weren't characterized very well. Note that *tank* is known to be a hard case. In a recent paper, Hearst obtained unsatisfactory results for *tank* with a method that worked well for other cases [Hearst 1991].

However, the relative success of a robust version of the nearest neighbor method mentioned above, shows that the data have the right structure although there may just not be enough instances for unsupervised learning to work. The following algorithm has an error rate of 9% (see Table 5). Rare senses that are unaccounted for are "think tank" and "tank top". (The numbers 168 and 8 were estimated from the disambiguation results of a quarter of the VEHICLE contexts.)

Disambiguation by nearest neighbor.

1. Compute the context vector of the test context;
2. Find the closest context vector in the training text that hasn't been used yet;
3. If two contexts of sense$_i$ have been found, disambiguate the polyseme as i; go to 2. otherwise.

Large scale applications

A problem in using sublexical analysis for large applications would be that one would need multiple vectors for every word in the lexicon, one for each polyseme. Given the large number of ambiguous words, this would blow up the amount of storage needed considerably. However, if a principal component analysis of a collocation matrix for the whole lexicon is computed, a uniform representation could be used for disambiguating all polysemes. Since the number of distinctions that can be made is exponential in the number of dimensions, one would hope that relatively low dimensional spaces are sufficient for disambiguation, although more than the up to 14 dimensions used in the above experiments would be needed.

The computational bottleneck is then the principal component analysis. Up to 30,000 × 50,000 matrices have been decomposed in information retrieval (Sue Dumais, p.c.). But since the time complexity of the algorithm is quadratic in the number of words and cubic in the number of dimensions [Deerwester *et al.* 1990], much larger matrices will become decomposable as

computer technology advances. The amount of space needed to store the collocation matrix depends on how many words cooccur with each other. The more low frequency words are included, the sparser the collocation matrix gets. The 4441 × 4441 matrix used for *tank* has more than 90% zeros if 1's, 2's and 3's are culled. Deerwester *et al.* report that omitting small counts has little effect on the decomposition. Since sparse collocation matrices can be compactly stored, demands on space thus don't seem to be a problem.

Conclusion

Let us take a look back at how disambiguation with sublexical representations works. The method relies on interpreting the words around an occurrence of the polyseme in question as cues for one sense or another. This is certainly no new idea. But the essentially automatic method presented here seems to be effective in integrating the constraints imposed by individual words. An example is the following context of *interest*. Although the training words *currency*, *exchange*, and *invest* seem to suggest the PERCENT sense, the program correctly disambiguates *interest* as CONCERN. (A stemmer deletes all suffixes, so that no information about the past tense morpheme in this context is available.) The 21 word window is delimited by ||'s.

> "If I had to take one currency, I'd pick the Swiss franc. It's gold-backed, and their interest rates are above 9 percent, which is very high for them || historically," said Drummond. He cautioned that he isn't particularly interested in the risks involved with playing currency exchanges. Investing in || stocks in a nation with a stronger currency than the dollar can partially offset stock price declines and boost price gains.

Disambiguation on the basis of sublexical representations has a number of advantages. It doesn't depend on the availability of thesauri and bilingual corpora; it seems to be efficient enough for large scale applications; and it is automatic, the only human intervention being the identification of the classes that have been found in unsupervised training. However, the disambiguation algorithm presented here doesn't use any information that is encoded in the order of words and ignores morphology and function words. Because such information is needed in many contexts, the best result obtained, 96% for *suit*, is probably an upper bound for performance. Future research has to be done on how the method can be extended to include a wider range of linguistic phenomena.

Acknowledgements. I'm indebted to Pratheep Balasingam, Doug Cutting, Marcus Grote, Marti Hearst, Martin Kay, David Magerman, Amir Najmi, Jan Pedersen, Scott Roy, Will Taylor, John Tukey, Tom Veatch and Hongyuan Zha for comments and help. Special thanks to Mark Liberman.

I'm grateful to Mike Berry for SVDPACK; to NASA and RIACS for AutoClass; to the San Diego Supercomputer Center for providing seed time; and to Xerox PARC for making corpora and corpus tools available to me.

References

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

Cheeseman, Peter, James Kelly, Matthew Self, John Stutz, Will Taylor, Don Freeman (1988): AutoClass: A Bayesian classification system. Proceedings of the Fifth International Conference on Machine Learning, University of Michigan, Ann Arbor.

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

van Gelder, Tim (1991): What is the "D" in "PDP"? A survey of the concept of distribution. In Ramsey, William M., Stephen P. Stich, David E. Rumelhart (1991): *Philosophy and connectionist theory.* L. Erlbaum Associates, Hillsdale NJ.

Golub, Gene H., Charles F. Van Loan (1989): *Matrix computations.* Second Edition. The Johns Hopkins University Press.

Hearst, Marti A. (1991): Noun homograph disambiguation using local context in large text corpora. In Proceedings of the Seventh Annual Conference of the UW Center for the New OED and Text Research.

Kawamoto, Alan H. (1988): Distributed representations of ambiguous words and their resolution in a connectionist network. In Small, Steven L., Garrison W. Cottrell, Michael K. Tanenhaus, Eds. *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence.* Morgan Kaufmann Inc., San Mateo CA.

Rumelhart, David E., James L. McClelland and the PDP Research Group (1986): *Parallel distributed processing: Explorations in the microstructure of cognition I.* MIT Press.

Salton, Gerard, Michael J. McGill (1983): *Introduction to modern information retrieval.* McGraw-Hill, New York.

Schütze, Hinrich (Forthcominga): Towards connectionist lexical semantics. In Roberta Corrigan, Greg Iverson, Susan D. Lima, Eds. *The Reality of Linguistic Rules.* John Benjamins.

Schütze, Hinrich (Forthcomingb): Dimensions of meaning. In *Proceedings of Supercomputing 92.*

Smolensky, Paul (1988): On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11,* 1–74.