In Proceedings of Supercomputing '92

Dimensions of Meaning

Hinrich Schütze
Center for the Study of Language and Information
Ventura Hall
Stanford, CA 94305-4115

Abstract

The representation of documents and queries as vectors in a high-dimensional space is well-established in information retrieval [1]. This paper proposes to represent the semantics of words and contexts in a text as vectors. The dimensions of the space are words and the initial vectors are determined by the words occurring close to the entity to be represented which implies that the space has several thousand dimensions (words). This makes the vector representations (which are dense) too cumbersome to use directly. Therefore, dimensionality reduction by means of a singular value decomposition is employed. The paper analyzes the structure of the vector representations and applies them to word sense disambiguation and thesaurus induction.

1 Introduction

In this paper a new representational scheme is introduced that tries to provide a basis for determining closeness in meaning. The approach is motivated by work on vector representations in information retrieval. In IR systems such as SMART and SIRE documents and queries are represented as vectors in term space [1]. The assumption is that two documents are similar to the extent that they contain the same words. An obvious extension of this methodology to the representation of contexts is to assign to each context the set of words that occur in close proximity, say in a window of fifty words. However, the same content can be expressed with very different words, so that in this simple scheme two contexts could have a similarity measure of 0 although they are very close in meaning.

The problem is that the absence or presence of a given word is very little information if we treat words as unanalyzed symbols or indices in term vectors. The lexical representations used for comparing contexts have to be enriched. The approach adopted here is to represent words as term vectors that reflect their pattern of usage in a large text corpus. Figure 1 shows how this can be done. The terms *cash* and *sport* are the dimensions of the space in which similarity is to be measured. The columns of the matrix represent the words *bank*, *interest*, and *finals*. Each entry in the matrix is a cooccurrence count. For instance, $a_{cash,bank} = 300$ encodes the fact that the words *cash* and *bank* cooccur 300 times in the (hypothetical) corpus. Cooccurrence can be defined with respect to windows of a given size or on the basis of sentence boundaries.

With cosine of the angle between the vectors as a measure, we get the following correlations for the three words in Figure 1: $\cos(bank, interest) = 0.94$, $\cos(interest, finals) = 0.92$, $\cos(bank, finals) = 0.74$. These numbers 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 graphical equivalent: Proximity in the multidimensional space corresponding to the collocation matrix. In Figure 2 *bank* and *finals* are not very close to each other, but both are close to the vector *interest* between them.

Now we are in a position to compute a representation of context that is more reliable than the bag-of-words method criticized above: The normalized average (or **centroid**) of the vectors of the words in a context can be seen as an approximation of its semantic content. If at least some of the words in the context are frequently used to describe what the current context is about then their vectors will pull the centroid

	bank	*interest*	*finals*
cash	300	210	133
sport	75	140	200

Figure 1: A collocation matrix.

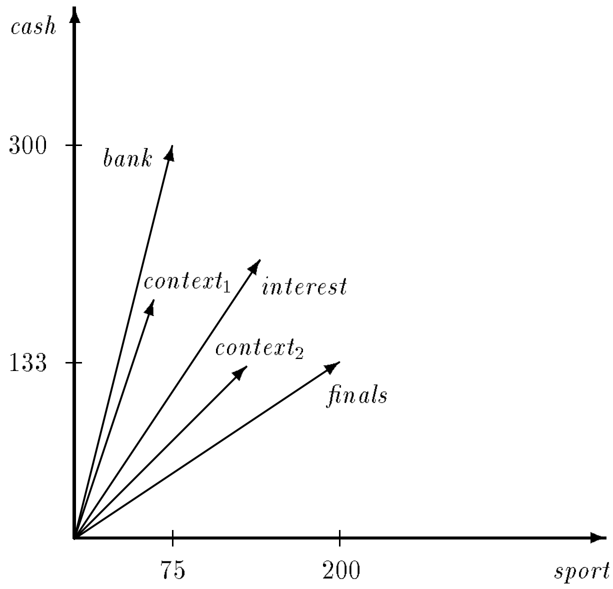

Figure 2: A vector model for context.

toward the direction of that topic or content. It is possible to defeat this scheme by describing a content exclusively using words that normally express unrelated thoughts. But such situations are expected to be rare.

Let us look at word sense disambiguation to see how this representation of context can be put to work. Consider the example of the word *interest*. Let PERCENT be the tag for uses of *interest* in the sense "charge on borrowed money" and CONCERN the tag for "a feeling that accompanies or causes special attention." Then the PERCENT sense will occur more often in contexts that score high on the *cash* dimension, and the CONCERN sense will occur more often in contexts that score high on the *sport* dimension. We can then disambiguate an occurrence of *interest* at a given position in the text by computing the context vector of that position and determining how close it is to the *cash* and *sport* dimensions of the space. Two such context vectors are depicted in Figure 2. Vector $context_1$ is closer to *cash*, so probably it is an occurrence of the PERCENT sense of *interest*. Vector $context_2$ is closer to *sport*, and it is most likely an occurrence of the CONCERN sense.

A space with only two dimensions, *cash* and *sport*, would be a rather impoverished representation. For better results, several thousand words should be considered. It is here that supercomputing becomes crucial. The collocation matrices usually have few zeros in them because large windows are used and the corpus has a size of more than 50 million words. As a result, almost every pair of words cooccurs. In practise, this means that all but about 10% of the cells are filled. For example, a typical 4000-by-4000 matrix had less than 10% zeros.

Any systematic work in this framework needs to use more efficient representations since vectors with several thousand components take up too much space and time in processing. Therefore, a dimensionality reduction by means of a singular value decomposition is performed. The algorithms from Mike Berry's SVDPACK were used in this paper, mainly the Lanczos algorithm LAS2.

As will be shown in section 4.3, the vector representations have the key properties of the distributed representations characteristic of parallel distributed processing [2]. They will therefore be referred to as **sublexical representations** in analogy to terms like "subsymbolic" and "subconceptual" in connectionism.

2 Word Sense Disambiguation

Word sense disambiguation is important for many areas of language processing. For instance, different senses of a word have different translations in foreign languages and they have to be rendered differently in a text-to-speech system.

The main problem in using sublexical representations for disambiguation is to find the directions in the space that correspond best to the various senses of an ambiguous word. One could imagine many labor-intensive ways of identifying such directions: for instance finding several dozen typical uses and computing their centroid. A less reliable, but automatic, method taken here is to cluster a training set of contexts, to assign senses to the clusters and to assign new occurrences the sense of the closest cluster. The clustering programs used are AutoClass [3] and Buckshot [4]. AutoClass is a Bayesian classification program based on the theory of finite mixtures. It determines the number of clusters automatically by imposing a penalty on each new cluster and thus counterbalancing the fact that more clusters will necessarily better account for the data. Due to the computational complexity of high-quality classification, buckshot, a more efficient, linear algorithm, was used for some of the large data sets shown in Table 1. Buckshot clusters n items by applying a quadratic high-quality clustering algorithm to a random sample of size $\sqrt{kn}$ (for some constant k) and extending this classification in linear time to the rest of the data set.

word	training set		test set						# contexts per sense				% correct			
	# months	# contexts	# months	# contexts	clustering	# classes	% rare senses	% major sense	1	2	3	sum	1	2	3	sum
capital/s	3	2000	1	200	A	2	5	66	127	64		191	96	92		95
interest/s	2	2955	1	501	A	3	15	68	291	165		456	94	92		93
motion/s	18	3101	1	200	B	2	0	54	107	93		200	92	91		92
plant/s	5	4132	1	502	A	13	14	66	283	188		471	94	88		92
ruling	18	5966	1	200	B	2	4	60	115	78		193	90	91		90
space	18	10126	1	200	B	10	0	59	118	82		200	89	90		90
suit/s	18	8206	1	498	B	2	18	54	220	189		409	94	95		95
tank/s	5	1780	1	336	A	8	16	80	226	56		282	97	85		95
train/s	18	4775	1	266	B	10	2	76	200	62		262	94	69		89
vessel/s	17	1701	2	144	B	7	10	58	76	23	22	130	93	91	86	92

Table 1: Ten disambiguation experiments.

word	sense	pos	definition
capital	1	N	stock of goods
	2	N	seat of government
interest	1	NV	a feeling of special attention
	2	N	a charge for borrowed money
motion	1	N	movement
	2	N	a proposal for action
plant	1	N	a factory
	2	NV	living being
ruling	1	NV	an authoritative decision
	2	V	to exert control, or influence
space	1	N	area, volume
	2	N	outer space
suit	1	N	an action or process in a court
	2	N	a set of garments
tank	1	N	a combat vehicle
	2	N	a receptacle for liquids
train	1	N	a line of railroad cars
	2	V	to teach
vessel	1	N	a ship or plane
	2	N	a blood vessel
	3	N	a hollow or concave utensil

Table 2: Definition of the senses in Table 1.

Table 1 summarizes the ten disambiguation experiments that have been conducted so far. The first column contains the word that is to be disambiguated. In two cases, inflected forms are excluded because they are not ambiguous. (*rulings* only has the "decision" sense, *spaces* cannot mean "outer space.") For all words, the training and test set were taken from the New York Times News Service. The training sets consisted of months from the period May 1989 through October 1990. The test set was in November 1990 except for *vessel* which was trained on June 1989 through October 1990 and tested on May 1989 and November 1990. Columns 3 and 5 show how often the ambiguous word occurred in test and training set. The column "clustering" has "A" for AutoClass and "B" for Buckshot. The next column gives the number of classes found by AutoClass or the number of classes requested for Buckshot. Usually, classifications with 2, 5, 7 and 10 classes were tried. The first successful trial is reported in the table.

Infrequent senses of the ambiguous words were excluded here. The percentage in column 8 ("% rare senses") indicates how many occurrences are not accounted for. It also includes repetitions of identical contexts for *tank*, *plant*, *interest*, *suit*, and *vessel*. For these words repeated contexts only count once.

The column "major sense" shows how dominant the major sense of the word is. For instance, 80% of the frequent uses of *tank* are "vehicle" uses, 20% "receptacle" uses.

Contexts in the test set were disambiguated according to the sense of the closest cluster. For instance, if

word	ten nearest neighbors
absolutely	absurd whatsoever totally exactly nothing does understood truly matter anyone
bottomed	dip copper drops topped slide trimmed slightly squeeze flat mix
captivating	shimmer stunningly superbly plucky witty melodrama fairy stoppers stylized tale
doghouse	dog porch crawling beside downstairs gazed alley sofa crawled upstairs
Makeup	repellent lotion glossy sunscreen Skin gel pokes hue mascara dyes
mediating	reconciliation negotiate cease conciliation peace EPLF talks immediate Nations OAS
keeping	hoping bring wiping could some would other here rest have
lithographs	drawings Picasso Dali sculptures Gauguin Monet paintings painters Degas artwork
pathogens	toxins bacteria organisms bacterial parasites humans microbial parasitic amino microbes
senses	grasp psyche truly clumsy naive innate awkward realm somehow instinct

Table 3: Ten randomly selected words and their nearest neighbors in sublexical space.

for a context of *tank* in the test text the closest cluster in the training set had been assigned the sense tag VEHICLE, then the context was disambiguated as belonging to that sense. The last six columns of Table 1 contain the absolute number of occurrences per sense and the percentage of correct disambiguation.

Table 2 glosses the major senses of the ten words. The column "pos" shows the part of speech of the word. Some senses can be realized as verbs or nouns. In general, verbs are harder to disambiguate than nouns, but as the results for *plant*, *interest*, *ruling* and *train* show, success in the 90% range is possible.

The senses that occurred in the New York Times and were excluded are "tank top" and "think tank" for *tank*; metaphorical senses such as "to plant a suction cup Garfield" and "the physical plant of a school" for *plant*; the "legal share" sense of *interest*; the adjectival and sports senses of *capital* ("capital punishment", "Washington Capitals"); the verbal sense of *suit* ("to be proper for"), the card game sense and "to follow suit"; "to rule out" for *ruling*; and "an orderly succession" and "drive-train" for *train*. Many of these senses occur in fixed expressions and are easy to filter out in preprocessing.

It is important to note that senses were assigned to classes on the basis of the training set. In the case of autoclass, only classes that had at most two errors among the first 10 members in the training corpus were assigned. In the case of buckshot, the majority of the first 10 or 20 members in the training corpus determined sense assignment. It is always possible to cluster the occurrences in the test set so finely that each cluster is homogeneous. In the extreme case, a classification with as many classes as items in the test set is guaranteed to be 100% correct. But since classes were assigned using the training set here, even a high number of classes seems unproblematic.

3 Word Space

When applied to word sense disambiguation, the information in sublexical space is reduced to a binary opposition: Is a particular context an instance of sense 1 or sense 2? But there is much more information in the sublexical representations of words: They can also be viewed as constituting a thesaurus by interpreting proximity in the space as a measure of semantic relatedness. Table 3 shows 10 out of 20 randomly selected words and their ten nearest neighbors. (The set of 20 words also contained proper names like "Chun" and trademarks like "Cheerios.") The neighbors are listed in the order of proximity to the head word.

The sample turns out to be representative: In general, the nearest neighbors of about 50% of the words in the space are as intuitive as the ones shown for *Makeup* or *lithographs*, but there is also a significant number of words like *keeping* that are not characterized well by their spatial neighborhood. The characterization is the better, the more clearly the set of typical topics of the word in question is delineated by other topics. *Makeup*, *mediating*, *lithographs*, and *pathogens* are all in topic areas with clear boundaries. Therefore, their nearest neighbors are other terms appropriate for describing this topic area. (EPLF stands for "Eritrean People's Liberation Front.") *absolutely*, *bottomed*, *captivating*, and *senses* have less clearly delineated topics. Therefore, their nearest neighbor sets contain some counterintuitive words. (*bottomed* is mainly used in financial contexts in the New York Times: "The market bottomed on April 27.") *keeping* can be used for almost any topic. For this reason, its nearest neighbors seem rather random. Finally, *doghouse* shows the limitations of the way the word space was computed. The key article in which the otherwise infrequent word *doghouse* occurs is a report on an exhibition of designer doghouses in New York: "New

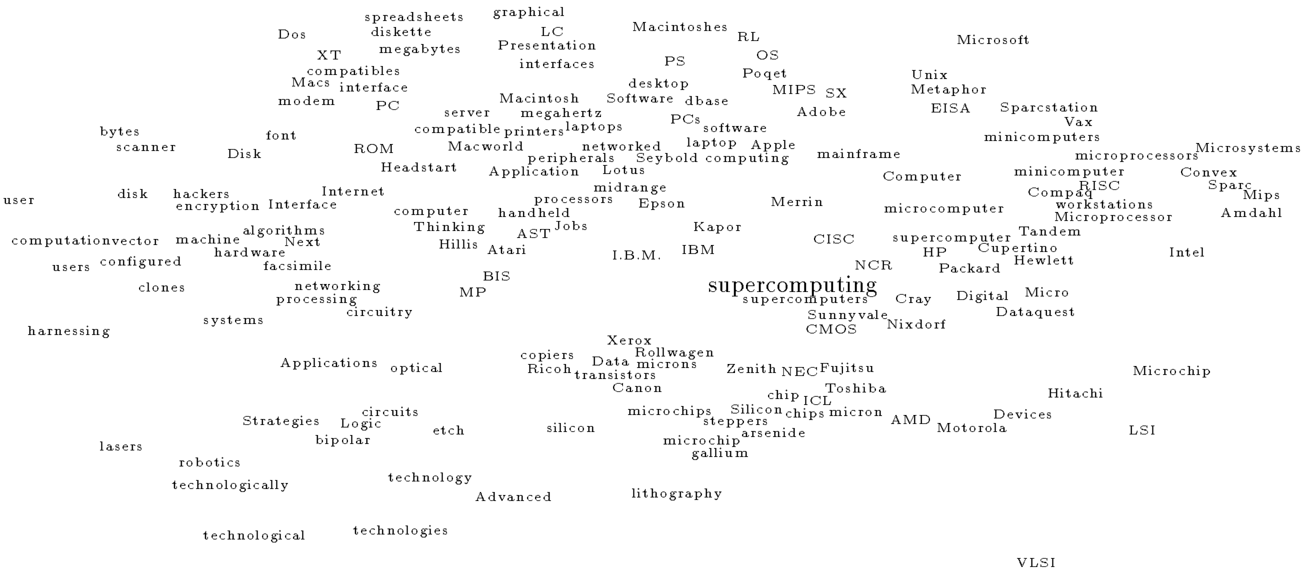

Figure 3: The semantic field of *supercomputing* in sublexical space.

York museum lands architects in 'The Doghouse' " (June 6, 1990). It contains many architectural terms and that is probably the reason that *porch*, *downstairs*, and *alley* ended up close to *doghouse*. Here, one article which was not very representative for the general usage of the word dominated the computation of its vector. This problem is likely to disappear if a larger corpus is used.

The word space and a classical thesaurus such as Roget's differ in that Roget's concentrates on synonyms or near synonyms whereas the nearest neighbors in sublexical space include mostly collocates, terms frequently used with the word in question. To illustrate, Roget's gives words like *dilemma*, *nut to crack*, and *Gordian knot* for *problem*. The most prominent neighbors in word space are *solve*, *recurring*, *panacea*, *beset*, *grips*. These are words that are important for using *problem* or for expressing a content similar to those that *problem* is used for, but they are not synonyms.

Table 3 is good for evaluating how well proximity in meaning and proximity in the space correlate, but it doesn't show the fine structure of the space at a particular location. The word *supercomputing* was chosen for a closer examination of a small region of the space. Figure 3 shows principal components 2 and 3 of the matrix of correlation coefficients of the 200 nearest neighbors of *supercomputing*.[1] The center of each word is plotted at the position determined by its projections on the second and third major axis of variation as computed by a principal component analysis. If several words have the same position, only one is shown. Some words were moved up to 4 points to make the figure more legible.

As can be seen from the figure, the representations are highly corpus-dependent. For instance, if the collocation matrix had been based on these proceedings, *minicomputers* would not be part of the semantic field of *supercomputing*. But in the New York Times, the topic of supercomputing is not clearly distinct from general computer science and the manufacturing of computing machinery. For many applications, it is an asset rather than a liability that the representations are corpus-dependent. Consider the example of information retrieval. If the New York Times is the document collection, it is likely to be used for financial queries. In such a context, fine distinctions between different branches of the computer industry don't seem necessary. On the other hand, a sublexical space based on a collection of articles devoted to research and development in computer science would have a much richer structure at the location of *supercomputing* and no articles on microcomputers would be retrieved by a query on supercomputing. Since little manual labor is involved in inducing the space, a representation that is finely tuned to the document collection in question could be a promising basis for many applications in information processing.

[1] The vectors are all very similar since they are from the same neighborhood. Principal component 1 captures that similarity, but no interesting variation in the data.

4 Analyzing Sublexical Space

How can the disambiguation results in Table 1 be improved? There are many parameters that had to be fixed rather arbitrarily and they may have gotten suboptimal settings. This section investigates three of them: the size of the window; the weighting of the dimensions of the space; and the selection of different sets of dimensions.

4.1 Window Size

In Table 1, window sizes of 1000 or 1200 characters were used for computing the context vector. It makes more sense to limit the window by the number of characters than by the number of words because few long words are as good as (or even better) than many short words which tend to be high-frequency function words. How does window size influence disambiguation performance? To answer this question one could cluster context sets that are computed with varying window sizes. However, there's some variability in the results of clustering and the best window size may yield mediocre disambiguation results by accident. An average over several clusterings could be taken, but that would be time-consuming. A deterministic, less expensive method is therefore needed.

Canonical Discriminant Analysis (CDA) or linear discrimination is such a method [5]. It finds the best weighting or linear combination of the dimensions of the space so that the ratio of the sum of between-group distances to the sum of the within-group distances is maximized. This task is slightly different from classification. It could be that, say, forty dimensions are sufficient for clustering, but that more are needed to tease the two words apart on a linear scale as CDA does. Conversely, even though giving large weights to few dimensions with very low values in the original space can result in a nice separation, a clustering procedure may not be able to take advantage of this situation because the distance measure is the cosine and it concentrates on dimensions with high values. So the results below have to be interpreted with some caution.

Linear discrimination is a supervised learning method: the items in the training set have to be labelled. Since labelling thousands of instances of an ambiguous word is not feasible, a simple trick was employed here. Instead of discriminating an ambiguous word for which the sense tags in the corpus are unknown, three artificially ambiguous words were created: *author/baby*, *giants/politicians*, and *train/tennis*. These pairs were selected because the words in each pair are comparable in frequency in the corpus and they are as distinct semantically as the different senses of ambiguous words like *suit* or *capital*. All six words are nouns because the meaning of verbs often depends on their arguments rather than on the general context. However, about twenty percent of the occurrences of *train* are verbs (see above). Table 4 lists the frequencies of the CDA words in training and test set.

pair	word	# contexts in training set Jun90–Oct90	# contexts in test set Nov90
pair 1	*author*	1552	312
	baby	1544	349
pair 2	*train*	1089	219
	tennis	1072	136
pair 3	*giants*	1544	707
	politicians	1530	364

Table 4: Frequency of the words used in CDA.

Figure 4 shows how generalization to the test set depends on the number of dimensions and the window size. The solid line is 1200 characters, the dense dotted line 1000 characters and the sparse dotted line 800 characters. Each point in the graph was computed as follows: For a given window size, a linear discrimination analysis was performed for the 3096 data points in the training set using the first n dimensions, where the value of n is indicated on the horizontal axis. The computed weighting was used to project the 3096 points onto one dimension. The optimal cutting point was determined. The projection and the cutting point were then applied to the test set. The graph shows how many contexts in the test set were discriminated correctly (in percent).

Apparently, 1000 characters is the ideal window size for discriminating *author/baby*. The results for *train/tennis* were similar in that 1000 characters seemed the optimal size most of the time although 800 and 1200 characters produced generalizations very close in quality for many dimensions. For *giants/politicians*, the graphs for the three window sizes were almost identical for most dimensions. This suggests that 1000 characters is a good window size for computing the context vector.

Figure 4 also suggests that using more than 97 dimensions could improve the disambiguation results. Unfortunately, only 97 dimensions were extracted when computing the singular value decomposition, so it could not be tested whether the curve keeps rising or flattens out fast beyond dimension 96. The discrimi-

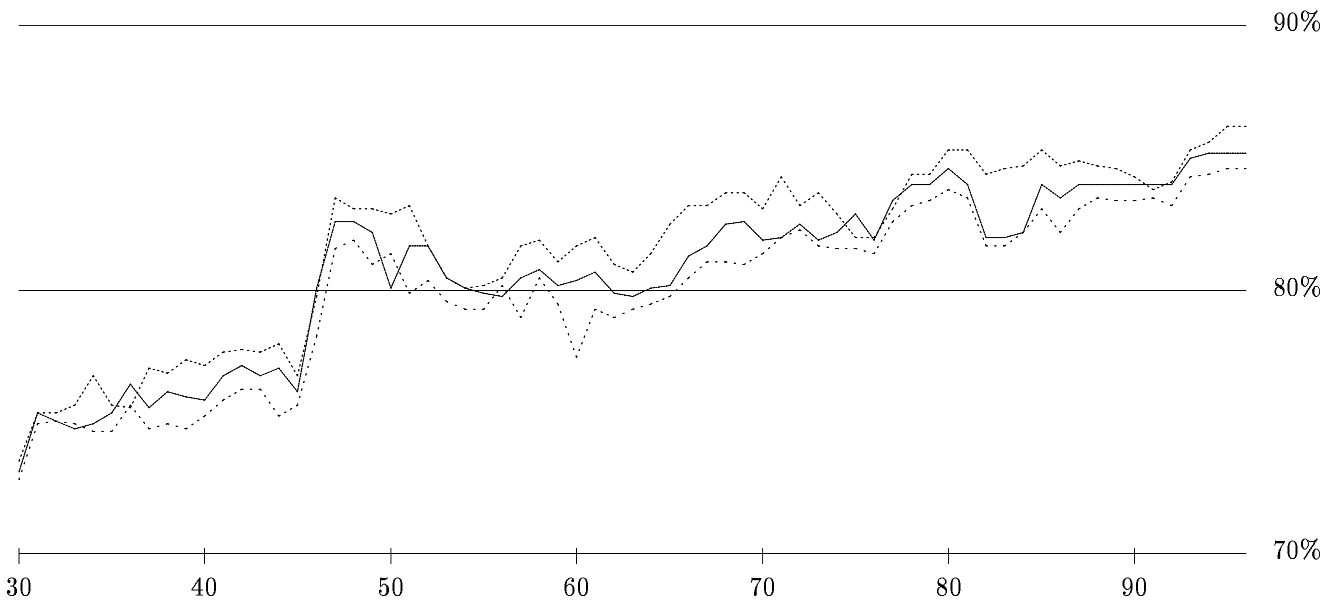

Figure 4: Discrimination of *author/baby* for different window sizes and dimension sets.

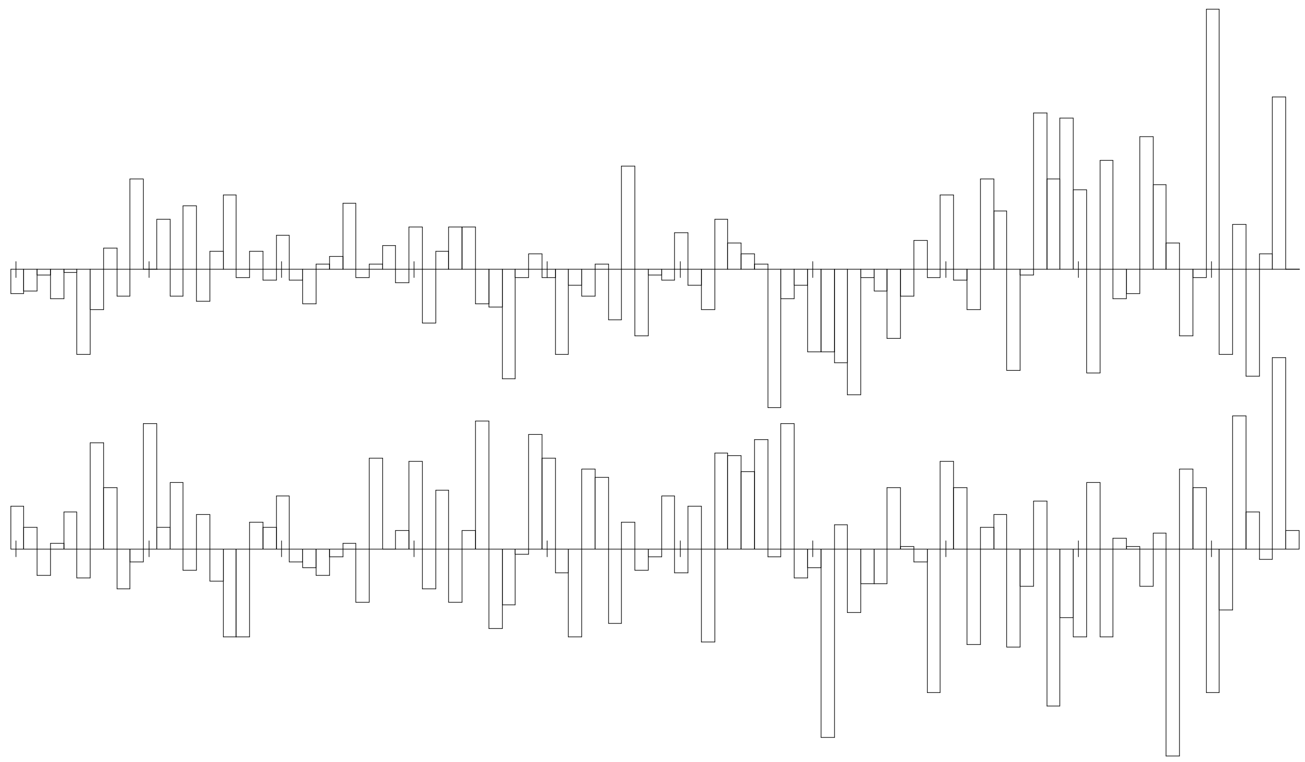

Figure 5: Optimal dimension weights for discriminating *author/baby* and *train/tennis*.

nation graph for *giants/politicians* has a very clear rising tendency, so an improvement in perfomance with more dimensions seems likely.

4.2 Dimension Weights

The second question is whether all dimensions are important for some distinctions or whether there are some that are never relevant. A preliminary answer can be found in Figure 5. It shows the optimal weights as computed by the CDA algorithm if all 97 dimensions are used. Dimensions 0, 10, 20, 30, 40, 50, 60, 70, 80, and 90 are marked on the horizontal axis in both figures. The height of each rectangle shows the relative weight of the corresponding dimension. The weightings were very stable when new dimensions were added, with each incoming dimension dampening the weights of the others without changing the "gestalt" of the weight graph.

Different weightings seem to be necessary for different word pairs. For instance, dimension 10 (the second marked dimension from the left) has weight zero for *author/baby*, but a high positive weight for *train/tennis*. Dimensions 70 and 80 (the second and third marked dimensions from the right) have weights with the same signs for *author/baby* and weights with different signs for *train/tennis*. So whereas high positive values on both 70 and 80 will strongly favor one sense over the other in discriminating *author/baby*, they cancel each other out for *train/tennis*. The optimal weights for *giants/politicians* display yet another pattern. This evidence indicates that different dimensions are important for different semantic distinctions and that all are potentially useful.

4.3 Distributed Representation

Three experiments were conducted to find out whether some groups of dimensions were more important than others. In general, a singular value decomposition yields a space in which the leading dimension is most important, the second dimension is the second most important etc. But this doesn't seem to be the case here as the disambiguation results in Table 5 show. Data set 1 (contexts of *suit*) in Table 5 was first classified using only dimensions 1 through 30. The error rate on the test set was 6%. Then a classification with the last 38 dimensions was computed, again yielding an error rate of 6%. Finally, all 97 dimensions except for the very first one were classified. Here, the error rate was 5%. This suggests that the vector representations are highly redundant and that

	dimensions used	error rate
data set 1	1–30	6%
	59–96	6%
	1–96	5%
data set 2	1,2,3,...,29,30	9%
	1,3,5,...,27,29	14%
	2,4,6,...,28,30	13%

Table 5: Sublexical representations are distributed.

the singular value decomposition computed here is different from other SVD applications in that the first one hundred dimensions are all equally meaningful for the disambiguation task. This hypothesis is confirmed by the classification of the second data set in Table 5 (also contexts of *suit*). Using all 30 dimensions, the error rate is 9%. Deleting either all even dimensions or all odd dimensions increases the error rate, but there's still enough information to find a classification of moderate quality.

It is also instructive to repeat the linear discrimination experiments for sets of final dimensions (as opposed to sets of initial dimensions as in Figure 4). There is evidence that the leading dimensions may actually have less relevant information in them than the immediately following ones. It was found that disambiguation on the basis of dimensions 51–96 attains almost optimal performance and adding dimensions 0–50 only leads to minor improvements. Only 30 final dimensions (67–96) are necessary for 80% correctness whereas almost 50 dimensions (0-48) are needed for the same level of performance in Figure 4. The curve for final dimensions also is initially much steeper than its counterpart in Figure 4. Further research is necessary to find out whether dimensions 100-200 are even better than 50–100.

5 Discussion

The approach to semantic representation proposed here bears some similarity to Latent Semantic Indexing (LSI) in information retrieval in that a singular value decomposition is used [6]. However, there is an important difference: In LSI, the main purpose of the space reduction is to improve the quality of the representations, thereby achieving better performance. The initial term-by-document matrix is noisy because it contains many small counts which are inherently unreliable. Using SVD as a smoothing technique removes this noise. However, the term-by-term matrices described in this paper are dense and mostly contain high counts, due to the size of the corpus. As mentioned above, the matrices typically contain less than 10% zeros, and more than 50% of the elements are greater than 100. So no smoothing is necessary to deal with the "bumpiness" of small counts.

The detection of term dependencies is another motivation for using SVD in LSI. Two documents may have a low similarity score in the original term space because they use different terms to express the same concept. For instance, document d_1 may use *coast*, where document d_2 uses *shoreline*. The truncated vectors of d_1 and d_2 computed by the singular value decomposition will be more similar than the original term vectors since *coast* and *shoreline* occur together in many documents. Loosely speaking, the singular value decomposition will assign them to the same prinicipal component. As a result, recall and precision improve when the truncated vectors from the SVD are used instead of the full term vectors. Again, the application of SVD presented here is different: The original vectors of *coast* and *shoreline* in the collocation matrix are already very similar since *coast* and *shoreline* coccur with the same words.

In order to test the prediction that noise and the detection of term dependencies do not play an important role in this application, one disambiguation experiment was repeated with the unreduced vectors. The columns of about one thousand words in the collocation matrix that cooccur with *interest* were normalized after the counts had been dampened by application of square root. 2954 context vectors of *interest* in June and July 1990 were computed by summing up the vectors of all words in a 51-word window around the occurrence of *interest*. This set of 2954 context vectors was then clustered into two classes using buckshot, and applied to the context vectors of the first 501 occurrences of *interest* in November 1990. Sense prediction was correct for 93% of the CONCERN contexts and 93% of the PERCENT contexts. This result is almost identical to the one with truncated vectors described above (94% for CONCERN, 92% for PERCENT). This indicates that, in contrast to LSI, the application of SVD does not influence performance in the case of sublexical representations.

On the other hand, the compact representation without loss of information that is made possible by the singular value decomposition is less important in LSI since document vectors in information retrieval are sparse and can be efficiently stored and processed in unreduced form. However, this compactness property of a principal component analysis is crucial for this paper. If 20,000 words were to be represented with 5000-component vectors each, a 100-megaword memory would be required, and any application program, for instance for word sense disambiguation, would be prohibitively slow. Sublexical representation therefore depends on high performance computing for any application that aims to be efficient enough for real world use.

Although the technical motivation for the dimensionality reduction is different, sublexical representation is very close to LSI as far as the interpretation of the dimensions of the reduced space is concerned. There is a long tradition in the social sciences of using principal component analyses to understand variation in large data sets. For instance, a survey of "The Elderly at Home" with 20 variables is subjected to a principal component analysis in [7]. The first eleven principal components are then interpreted as corresponding to elderly people living alone vs. those that share accommodations with others etc. The approach taken here follows LSI in that there is no interest in interpreting the dimensions, gaining additional insights as to the structure of the data, or rotating the space in order to position the axes in an intuitive way. All directions in the space are treated equally. The only important information is the measure of similarity that can be obtained for any two words or contexts by computing their correlation coefficient.

The disambiguation results achieved here compare favorably with those reported for other approaches. For instance, the methods in [8, 9] perform slightly better than the average of 92% in Table 1. However, they rely on thesauri and bilingual corpora. For many technical domains and foreign languages, thesauri or bilingual corpora are not available. Word sense disambiguation on the basis of sublexical representation only needs raw text as input, so there is virtually no limitation to its application.

Representations that are derived by means of a dimensionality reduction differ from other statistical approaches in that a small number of parameters (on the order of a few thousand) is estimated. Trigram-based models such as the one presented in [10] have to estimate millions or even billions of parameters. Even the largest corpus is not sufficient to estimate such a large number of parameters reliably. In contrast, a couple of hundred principal components can be easily justified with a corpus of 10 million words, resulting in robust estimates of statistical parameters.

The approach to word sense disambiguation proposed here is also different from knowledge-intensive methods. In classical AI, word sense disambiguation is based on knowledge representation and logical inference. The challenge is to encode all items of knowledge that may be relevant for tasks like disambiguation and to integrate them into a system that will respond appropriately. This goal has not been achieved yet and systems like Cyc [11] still seem far from coming close to it. The application of principal component analyses in this paper can be seen as a tool for integrating a large number of constraints, each word imposing a constraint as to which sense is more likely in its neighborhood.

6 Conclusion

The basic idea of this paper is to take the notion of semantic similarity seriously. In order for the "dimensions of meaning" and the vector representations of words to reflect closeness in meaning faithfully, a global optimization of cooccurrence constraints is necessary, an operation so complex that only a supercomputer can perform it. Semantic similarity underlies many processes in linguistics (for instance metonymy: the replacement of the name of one thing by a closely related one) and psychology (for instance priming: after the presentation of a concept to a subject, reaction times are short for semantically related terms and long for unrelated ones). A host of recent papers on mutual information (for instance [12]) is witness to its importance in computational linguistics and lexicography.

Still, even if semantic similarity is important, a more intricate set of lexical relations is needed for more ambitious natural language processing and linguistics, relations such as hyponymy or antonomy. These relations cannot be read off sublexical space as easily as semantic relatedness, but the space could be the basis of representation for semantic theories dealing with them. Representations and processes tend to go hand in hand; the way knowledge is represented largely fixes appropriate processes and vice versa. The novel approach to semantic representation presented here, an approach made possible by the availability of supercomputers to linguistic research, may thus lead to theories of semantics that look very different from today's.

Acknowledgements

I'm indebted to Ken Church, Patrick Hanks, John Maxwell, and John Tukey for comments, and to Martin Kay and Jan Pedersen for discussions and help.

I'm grateful to Mike Berry for SVDPACK; to NASA for AutoClass; to SDSC for computing time; and to Xerox PARC for corpora and corpus tools.

References

[1] G. Salton and M. J. McGill, *Introduction to modern information retrieval.* New York: McGraw-Hill, 1983.

[2] D. E. Rumelhart, J. L. McClelland, and the PDP research group, *Parallel Distributed Processing.* Cambridge MA: The MIT Press, 1986.

[3] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, "AutoClass: A Bayesian classification system," in *Proceedings of the Fifth International Conference on Machine Learning*, 1988.

[4] D. Cutting, D. Karger, J. Pedersen, and J. Tukey, "Scatter-gather: A cluster-based approach to browsing large document collections," in *Proceedings of SIGIR'92*, 1992.

[5] R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations.* New York: John Wiley & Sons, 1977.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[7] I. T. Jolliffe, *Principal Component Analysis.* New York: Springer-Verlag, 1986.

[8] D. Yarowsky, "Word-sense disambiguation using statistical models of Roget's categories trained on large corpora," in *Proceedings of Coling-92*, 1992.

[9] W. A. Gale, K. W. Church, and D. Yarowsky, "Using bilingual materials to develop word sense disambiguation methods," in *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, 1992.

[10] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "Word-sense disambiguation using statistical methods," in *Proceedings of ACL 29*, 1991.

[11] D. B. Lenat and R. V. Guha, *Building Large Knowledge-Based Systems.* Reading MA: Addison-Wesley, 1989.

[12] K. W. Church and P. Hanks, "Word association norms, mutual information and lexicography," in *Proceedings of ACL 27*, 1989.