Two-dimensional clusters in grammatical relations

Mats Rooth

1 Introduction

If someone asked me to explain the word "product" as used below, I might say that it refers to things like drugs, cars, and software which are developed, produced, sold, and used.

(1) The company announced a new product.

While it is drugs, cars and software which are serving as examples of products, the list of verbs has a role as well. If I merely said that products were things like drugs, cars, and software, someone would hardly be in a position to say whether toothpaste counts as a product. And a definition of "product" as simply "something which is produced" gets a more general sense of the word, including things such as toxic waste produced by an industrial process, and carbon dioxide produced by human respiration.

The explanation of "product" by multiplied example involves the implicit claim that drugs, cars, and software are all developed, and all produced, all sold, and all used. Not only is this true, but in a large enough text corpus we can find examples examples of drugs, cars, and software being *described* as being developed, produced, sold and used, in all combinations:

(2) a. 19516892 Their goal is to develop new drugs to compete in world markets .

b. 44435981 Genzyme sells specialty chemicals used by other companies to produce drugs , diagnostic tests and other products and performs contract research for drug companies .

c. 44222175 " We absolutely must be in the U.S. market by 1992 , " he says , " to sell the new drugs our research labs will be producing . "

d. 35971334 Milton Bass , a New York attorney for Zenith , asserted that the appeals court's ruling will spur civil suits against other drug companies that have " conducted campaigns to frighten doctors and others not to use generic drugs . "

(3) a. 56955552 The Europeans' problem: the huge sums it takes to develop new cars and bring them to market .

b. 47029320 The eight major auto makers didn't produce any cars last week because of the holidays .

c. 17150902 " This year , for the first time , we've had to work to sell the cars , " says Michael J. Jackson , a Saab dealer in suburban Washington , D.C .

d. 61132866 – A halt to imports of sedans , plus a requirement that top officials use only Chinese-made cars .

(4) a. 12402911 " This settlement paves the way for the competition to develop software in the operating system arena . "

b. 53998616 He guessed it will take Microsoft six to 12 months to produce the software based on the Sybase technology .

c. 34138820 The company operates retail stores that sell computer software .

d. 17518485 They are writing a book to show campaign managers how to use Lotus 1-2-3 software to analyze local voting habits .

These examples are from a corpus of Wall Street Journal newspaper stories, containing roughly 60 million word-like tokens (Liberman [Liberman, 1992]). The number at the beginning of an example indicates its position in the corpus.

This combinatory pattern evident in these sentences is of independent interest, because it suggests a simple way of capturing selectional patterns relating verbs to their objects, or lexical pairs participating in other grammatical relations. In a number of problems arising in computational linguistics, we need to be able to decide whether a given phrase is an appropriate filler for a grammatical relation assigned by another word. This often comes down to a question of semantic compatibility between the head of the filler phrase and the semantic role associated with the grammatical relation. For instance, in the sentence below, *resulted* and *approved* both have morphological readings as both tensed verbs and past participles.

(5) The charge resulted from a settlement approved yesterday.

The morphological indeterminacy results in syntactic ambiguity. In the first structure in the first row of table 1 , *charge* is the subject of *resulted* and *approved* a postmodifier of *settlement*. In the second analysis, *resulted* is a postmodifier of *charge*, and *charge* is the subject of *approved*. Deciding between the syntactic analyses below comes down, at least in part, to a matter of semantic selection. In choosing an analysis, it would be useful to know (among other things) whether *settlement* is a good object for *approve* (it is) and whether *charge* is a good subject for *approve* (it is not). (Note that in terms of the underlying role assigned, the postmodified noun phrases are equivalent to objects of their modifiers.) The sentence below has the same ambiguity,

Table 1: First row: correxts and incorrect syntactic analyses for sentence (x). Second row: incorrect and correct syntactic analyses for sentence (y).

and in this case it is the second syntactic analyis which is correct.

(6) Private-sector union contracts signed in the third quarter granted slightly lower wage increases.

Suppose that we wanted to attack such problems using information from a training corpus. Since thousands of verbs and nouns are involved, we can not conclude that a given verb-object pair is impossible, simply because we can not find it in the corpus. Take for instance a slightly more uncommon product verb such as *export*, and a more uncommon product noun, such as *engine*. Although the verb-object pair *export engine* is intuitively plausible, it was not detected at all by the method described below in a six million word training corpsus. A selection model which generalizes among words has the potential of solving the problem, since while the verb export *export* does not occur with *engine*, it occurs with other product nouns, and *engine* occurs with other product verbs.

The purpose of this paper is to develop a mathematical and computational model which captures the notion of a selectional dependency between a set of verbs and a set of nouns, or more generally two sets of words participating in a

	asset	average	bit	bond	cent	cost	debt	dividend	foot	interest	mark	pence	point	price	rate	rating	security	share	stake	stock	tax	unit	value	yen
acquire	16		1			1	2			35							5	77	87	29		19		
boost		1		1		2	1	18		1	1		1	39	21	5		28	59	10	1		19	1
buy	16	2	2	48			2			53	1						36	348	107	190		29		2
climb					8	1					2	10	13	1										
cut						104	10	11					1	66	64	11			5	2	30		5	
decline		2	3		18	2	1						13	9	16			1		3				
drop	1	1	1		19				2		1	2	30	9	6	5								7
dump	1			3													2	10		10				
fall		2	1		132				1		2	14	171	38	33					1				28
gain					20					3	2	11	62		9			25	4		25			28
hold	18			7	1				1	22				3			5	68	121	30		3	1	
increase	6	3			2	25	5	26		8		1	3	26	36	2	1	36	75	2	11		16	
jump		2			3						2	8	9											
lower						20	2			1				23	83	55			16	1	2		4	
plunge					3						2		14	4						2				
purchase	8			5			2		1	17							6	95	24	20		6		
push	1	2		2		1				1			3	44	20		1	4		16	1	1	2	1
raise						23	5	28		8	5			131	149	26		5	74		46		11	1
reduce	9	1			1	76	105	3		5			1	22	55	8		9	41	2	26		21	
retain	1							1		13								17	21	1		3	1	
rise		13	9		136	18			2	2	3	52	125	18	19			1		1		1	2	22
sell	114	2	1	40			6			72	2		1	12	8		48	243	144	149		104		2
slash						17	4	9						20	6				1	1	3		3	
trade	1	1		2	2	2								9			7	22	2	37		5		1

Table 2: Frequency counts for 24 verbs and 24 object heads.

grammatical relation. Section 2 describes a simple categorical selection model, which in section 3 is given a probabilistic twist. Locally optimal probabilistic models can be generated by an incremental procedure similar to Baum-Welch reestimation of hidden Markov models. In section 4 we look at results for a reasonably large sample of verbs an nouns. Section 5 discusses applications of the probabilistic model, giving preliminary results for a parsing problem.

2 Categorical selection types

Table (2) is a matrix of frequency counts for verb-object occurrences of twenty-four verbs and twenty-four nouns. The table is a sub-part of a verb-object matrix derived from an approximately 6 million word sample of the Wall Street Journal, parsed by Donald Hindle with his Fidditch parser.[1] I extracted verb-noun pairs with a Lisp program from list representations of parses, and mapped them to uninflected forms using a full form word list. The resulting list contained 84182 non-zero frequency counts. The frequency matrix was reduced by eliminating rows (corresponding to verbs) with fewer than five non-zero entries, and subsequently eliminating columns with fewer than five non-zero entries. This gave a matrix indexed by 992 verbs and 1027 nouns, containing 55251 non-zero entries. The verbs and nouns in the 24×24 sub-table were selected by hand for illustrative purposes.

Re-arranging the rows and colums in the small table brings out a dependency between the rows and columns. In table 2, most of the non-zero entries

[1]The parser is described in [Hindle, 1983] and [Hindle, 1993]. Similar verb-object data is discussed in [Church et al., 199x].

	asset	bond	interest	security	share	stake	stock	unit	average	bit	cent	foot	mark	pence	point	yen	cost	debt	dividend	price	rate	rating	tax	value
acquire	16		35	5	77	87	29	19		1							1	2						
buy	16	48	53	36	348	107	190	29	2	2			1			2		2						
dump	1	3		2	10		10																	
hold	18	7	22	5	68	121	30	3			1	1								3				1
purchase	8	5	17	6	95	24	20	6				1						2						
retain	1		13		17	21	1	3											1					1
sell	114	40	72	48	243	144	149	104	2	1			2		1	2		6		12	8			
trade	1	2		7	22	2	37	5	1		2					1	2			9				
climb											8		2	10	13		1			1				
decline					1		3		2	3	18				13		2	1		9	16			
drop	1								1	1	19	2	1	2	30	7				9	6	5		
fall							1		2	1	132	1	2	14	171	28				38	33			
gain			3		25	4					20		2	11	62	28					9		25	
jump									2		3		2	8	9									
rise			2		1		1	1	13	9	136	2	3	52	125	22	18			18	19			2
plunge							2				3		2		14					4				
boost		1	1		28	59	10		1				1		1	1	2	1	18	39	21	5	1	19
cut						5	2								1		104	10	11	66	64	11	30	5
increase	6		8	1	36	75	2		3		2			1	3		25	5	26	26	36	2	11	16
lower			1			16	1										20	2		23	83	55	2	4
push	1	2	1	1	4		16	1	2						3	1	1			44	20		1	2
raise			8		5	74							5			1	23	5	28	131	149	26	46	11
reduce	9		5		9	41	2		1		1				1		76	105	3	22	55	8	26	21
slash						1	1										17	4	9	20	6		3	3

Table 3: The same counts in another order.

are in the three blocks on the diagonal. We can think of the block organization as capturing three semantic selectional types within this part of the verb-object grammatical relation. The first block corresponds to a notion of exchange of financial instruments or ownership interests. The second block involves measurement of a scalar motion by some dimensioned quantity, and the third block involves change in some scalar quantity, such as a stock price. To represent the blocks, we need not re-order the matrix: we can equate a block with a pair of a subset of the verb set and and a subset of the noun set. A closer examination shows that it is not reasonable to insist that the noun sets for different blocks be disjoint. The noun *stake* occurs freqently with the verbs of the third block (e.g. *boost, increase, raise, and reduce*), as well as with the verbs of the first (e.g. *acquire, buy, sell, and trade*). Here are some example sentences:

(7) a. 1727999 Most of Japan's big computer companies hold a 1% stake in Ascii, and Mitsui & Co., one of Japan's largest trading companies, plans to boost its stake to 5%.

b. 1266211 Though Koito is resisting, and Mr. Pickens has announced plans to increase his stake to 26%, he maintains "there's nothing hostile" about his investment.

c. 1690314 Ford will raise its Jaguar stake to the maximum 15% after the 30-day waiting period expires, the Ford executive said.

d. 2081122 The sales reduce the group's stake to 740,500 shares.

(8) a. 5786677 Galesi Group said it is seeking to acquire Lone Star Technologies Inc.'s stake in American Federal Bank for $58 million.

b. 1024554 The Italian financier is close to announcing that he's setting up a holding company to buy controlling stakes in Hungarian companies, according to Hungarian sources.

c. 2668376 Domino's Pizza owner Thomas Monaghan may sell his 97% stake in the chain, which is the nation's largest pizza delivery company.

d. 4287292 One possibility would be for the U.S. group, formally called Newgateway PLC, to trade its troublesome stake to Isosceles for certain Gateway assets.

The reason for the overlap is that a stake (say in a company) is both something which can be bought or sold, and a scalar quantity which can be increased or decreased.

The verb *increase* is in principle a symmetric example, though this is not really obvious in these data. It occurs both with objects denoting a changing scalar quantity, and with a objects (or pseodo-objects) measuring that change:

(9) a. 6664778 Jack W. Forrest, Environmental Systems president and chief executive officer, said the change increases the company's effective tax rate to about 35% from 20%.

b. 1891048 If Mr. Wanniski's theory is right, that a tax cut increases the value of capital assets held by owners, wouldn't that increase also represent a rather dramatic and totally unjustified inflation of those values?

c. 4102768 Georgia Gulf said it increased the exercise price of the rights to $120 from $50.

d. 3578950 "It has increased the level of caution in the market," said the Chicago trader.

(10)a. 3337850 Economists said the August civilian unemployment rate will have increased 0.1 percentage point to 5.3%.

b. 3854070 Unleaded-gasoline futures were mixed, although the September contract increased 0.22 cent to settle at 54.15 cents a gallon.

In the latter group, the changing scalar quantity is realized as subject.

Given two vocabularies V and N, we define a selection type as a pair $\langle V', N' \rangle$, where $V' \subseteq V$ and $N' \subseteq N$. A selectional model is simply a set of selectional types, and given what was said above, we should not impose any requirement of non-overlap for the noun sets or verb sets of a selection model.

A reasonable selection model for the 24×24 frequency matrix is:

$$\left\{ \begin{array}{l} \left\langle \left\{ \begin{array}{l} \text{acquire buy} \\ \text{dump hold} \\ \text{purchase retain} \\ \text{sell} \end{array} \right\}, \left\{ \begin{array}{l} \text{asset bond} \\ \text{interest security} \\ \text{share stake} \\ \text{stock unit} \end{array} \right\} \right\rangle \\ \left\langle \left\{ \begin{array}{l} \text{climb decline} \\ \text{drop fall gain} \\ \text{jump rise} \\ \text{plunge increase} \end{array} \right\}, \left\{ \begin{array}{l} \text{average bit cent} \\ \text{foot mark pence} \\ \text{point yen} \end{array} \right\} \right\rangle \\ \left\langle \left\{ \begin{array}{l} \text{boost cut} \\ \text{increase lower} \\ \text{push raise} \\ \text{reduce slash} \end{array} \right\}, \left\{ \begin{array}{l} \text{cost debt} \\ \text{dividend price} \\ \text{rate rating tax} \\ \text{value share} \\ \text{stake stock} \end{array} \right\} \right\rangle \end{array} \right\}$$

3 Probabilistic selection types

The above construction, because it is categorical, encodes no information about the relative freqeuncy of, for instance, *buy* and *dump* as verbal realizations of the first selectional type. In applications, having access to graded information is useful, in that it allows the large numbers of analyses — such as syntactic parses — to be ranked. Futhermore, once we get beyond simple examples, it is not clear that membership in selectional patterns should be considered discrete.[2] Among ways of introducing graded distinctions, probabilistic models are appealing, because they have the potential of telling us, in complex situations, how a number of graded distinctions are to be combined. The simple recipe for turning a categorical model into a probability model is replace characteristic functions of sets with probability distributions. In the present case, we redefine a selection type as a pair of discrete distributions, one on the verbs and one on the nouns:[3]

$$\langle \lambda v p_v^\tau, \lambda n p_n^\tau \rangle \quad \sum_{v \epsilon V} p_v^\tau = 1 \quad \sum_{n \epsilon N} p_n^\tau = 1$$

The function $\lambda v p_v^\tau$ maps a verb to a number in the interval $[0, 1]$, meeting the constraint that the set of verb probalities sums to one, and similarly for the nouns. In order to use this notation, we must assume that the verb, noun

[2] Furthermore, I did not say what makes one categorical selectional model better than another, and how they are to be discovered computationally. I have experimented with an incremental search for selection models, using a Solomonoff-Kolmogoroff-Chaitin measure to evaluate the combination of the complexity of the model with the complexity of describing the data matrix given the model. I will not describe this method here, since the search was computationally expensive, and the results only moderately encouraging.

[3] $\lambda v p_v^\tau$ generates a probability distribution $\lambda X \sum_{v \epsilon X} p_v^\tau$ measuring subsets of V. In the text, I surpress the distinction between discrete probability distributions and their generators.

type 1 .204				type 2 .315				type 3 .481			
fall	.344	point	.364	raise	.265	rate	.238	sell	.319	share	.340
rise	.343	cent	.281	reduce	.189	price	.207	buy	.288	stake	.198
gain	.128	pence	.080	cut	.162	cost	.143	hold	.095	stock	.172
drop	.059	yen	.071	lower	.109	stake	.108	acquire	.093	interest	.078
decline	.038	price	.060	increase	.100	debt	.071	purchase	.063	asset	.065
climb	.028	rate	.059	boost	.073	tax	.063	increase	.030	unit	.059
jump	.020	tax	.021	push	.036	rating	.059	trade	.027	security	.038
plunge	.019	average	.017	slash	.034	dividend	.050	boost	.024	bond	.037
increase	.006	cost	.016	sell	.010	value	.044	retain	.019	debt	.004
push	.005	bit	.011	decline	.009	interest	.006	gain	.011	price	.002
trade	.003	mark	.011	drop	.006	stock	.003	dump	.009	average	.002
reduce	.002	foot	.004	trade	.005	mark	.003	push	.009	yen	.002
boost	.001	stock	.002	hold	.002	average	.002	reduce	.008	bit	.001
sell	.001	value	.002	retain	.001	asset	.002	raise	.003	mark	.001
cut	.001	interest	.002	acquire	.001	yen	.001	decline	.001	foot	.001
hold	.001	unit	.001	gain	.000	share	.000	rise	.001	value	.000
raise	.000	asset	.000	rise	.000	point	.000	plunge	.001	rate	.000
buy	.000	share	.000	plunge	.000	cent	.000	drop	.000	point	.000
slash	.000	rating	.000	fall	.000	pence	.000	slash	.000	cost	.000
lower	.000	stake	.000	climb	.000	bond	.000	lower	.000	cent	.000
acquire	.000	debt	.000	purchase	.000	security	.000	cut	.000	dividend	.000
purchase	.000	dividend	.000	buy	.000	bit	.000	fall	.000	tax	.000
retain	.000	security	.000	jump	.000	unit	.000	jump	.000	pence	.000
dump	.000	bond	.000	dump	.000	foot	.000	climb	.000	rating	.000

Table 4: Parameters of a selection model.

and type sets are (or have been rendered) disjoint: we do not want to identify the verb probability $p^\tau_{\text{increase/V}}$ with the noun probability $p^\tau_{\text{increase/N}}$. We also add a probability distribution over the types. Using an initial segment of the natural numbers to index the types, a probabilistic selection model for V and N with k types then consists of a probability distribution $\lambda\tau p_\tau$ over the set of integers $\{1, ..., k\}$ $(= T)$, and for each type τ in $\{1, ..., k\}$, a pair of probability distributions $\langle \lambda v p^\tau_v, \lambda n p^\tau_n \rangle$, as described above.

Derivatively, for any type τ we construct a probability distribution on $V \times N$ as a product :

$$p^\tau_{v,n} = p^\tau_v p^\tau_n$$

We constuct a probability distribution on $T \times V \times N$ as a disjoint union of these products:

$$p_{\tau,v,n} = p_\tau p^\tau_v p^\tau_n$$

Such a model can by used to assign probabilities to verb-object pairs. In the probility space just defined, if a verb-noun pair is generated, it is generated in some type, and we obtain the probability for the verb-noun pair by summing over the types:

$$p_{v,n} = \sum_\tau p_{\tau,v,n} = \sum_\tau p_\tau p^\tau_v p^\tau_n$$

In section 5, such probabilities are used to compare two grammatical analyses. Table 3 gives the parameters of a selection model of order three for the 24×24 data.[4] Notice that the noun *stake* is ranked high in both the second and third types.

[4] Or rather, as one can discover by summing the first column of numbers, the approximate parameters.

type 1 .204				type 2 .315				type 3 .481			
fall	.344	point	.364	raise	.265	rate	.238	sell	.319	share	.340
rise	.343	cent	.281	reduce	.189	price	.207	buy	.288	stake	.198
gain	.128	pence	.080	cut	.162	cost	.143	hold	.095	stock	.172
drop	.059	yen	.071	lower	.109	debt	.071	acquire	.093	interest	.078
decline	.038	average	.017	increase	.100	tax	.063	purchase	.063	asset	.065
climb	.028	bit	.011	boost	.073	rating	.059	trade	.027	unit	.059
jump	.020	mark	.011	push	.036	dividend	.050	retain	.019	security	.038
plunge	.019	foot	.004	slash	.034	value	.044	dump	.009	bond	.037

Table 5: The same model, with verbs and nouns represented only where they are most likely to be generated.

Estimating a model

Given a selection model and observed verb-object pair $\langle v, n \rangle$, the probability that it is generated in type τ is:

$$\frac{p_{\tau,v,n}}{p_{v,n}}$$

Multiplying by the frequency $f_{v,n}$, we obtain the expected number of occurrences of the event $\langle \tau, v, n \rangle$ given the observed frequency and the model:

$$e_{\tau,v,n} = f_{v,n} \frac{p_{\tau,v,n}}{p_{v,n}}$$

This forms a basis for re-estimating the probability parameters:

$$e_{\tau,v} = \sum_n e_{\tau,v,n} \quad e_{\tau,n} = \sum_v e_{\tau,v,n} \quad e_\tau = \sum_{v,n} e_{\tau,v,n}$$

$$q_v^\tau = \frac{e_{\tau,v}}{e^\tau} \qquad q_n^\tau = \frac{e_{\tau,n}}{e^\tau} \qquad q_\tau = \frac{e^\tau}{\sum_{n,v} f_{n,v}}$$

The probabilities q, the parameters of the new model, are computed as relative frequencies of expected numbers of events, as determined by the old model. For instance, the probability of the verb *fall* within type 1 would the expected number of occurences of *fall* in that type (given the data and model), divided by the expected number of occurrences of that type of verb-object pair.

The formulas are similar to the Baum-Welch re-estimation formulas for hidden Markov models (Baum 1972).[5] Adapting Baum's result for HMMs, it can be shown that an iterative re-estimatiom of parameters produces local improvements in the probability of the observed data given the model, and converges to a local maximum of this probability. Table 3 was derived in one hundred iterations starting from a random state, using the frequency data in

[5]Selection models as described here can be viewed as zero-order HMMs, augmented with a second surface vocabulary and associated emission probabilities. That is, given a state (or type), two surface symbols are independently generated. Furthermore, as used here, the types are hidden in the sense that they are given no prior interpretation, and are not observed in the data.

Table 2. Table 3 is a different way of looking at the same model: each verb or noun is is shown only in the type where it is most likely to be generated, i.e. where $p_\tau p_v^\tau$ (or $p_\tau p_n^\tau$) is maximal. This representation reconstructs the three eight-by-eight boxes of table 2.

4 Results for a larger data matrix

Storage requirements for the estimation algorithm are modest. There are $|T||V| + |T||N| + |T|$ probability parameters. The re-estimation formulas sum over non-zero frequencies, and the expectations can be computed by summing iteratively over such frequencies. In each step, a frequency $f_{v,n}$ is apportioned among the types according to the ratio $\frac{p_{\tau,v,n}}{p_{v,n}}$, and the portion for a type τ is added to running subtotals of $e_{\tau,v}$ and $e_{\tau,n}$. This procedure requires intermediate storage of the same size as the probability model. Since no random access to the frequencies is required there is no need to represent a $V \times N$ matrix. (This might turn out to be useful. In a larger data matrix based on 60 million words of text, about 3500 verb roots occur with five or more different nouns, and about 7500 noun roots, not counting proper nouns or numbers, occur with five or more verbs. One of these numbers would be multiplied if verbs with complements accompanied by prepositions were included as separate entries.)

The algorithm was implemented in Common Lisp. Tables 4 and 4 give the most probable nouns and verbs in each type of a probabiity model for the 992×1027 matrix with thirty-two types, obtained with four hundred iterations. The three blocks of 2 are represented here: type 3 is the product type (e.g. *develop software*), changing dimensioned objects (*raise price*) are in type 8, and scalar increments (*rise cent*) in type 26. Type 30 is a related one where the object typically denotes a scalar motion event, such as a decline or an increase. The nouns of types 19 and 29 primarily name people. In the nouns, the split seems to amount roughly to a distinction between powerful, active people (executives, lawyers, officials) and weak or passive ones (workers, clients, and shareholders). Looking at the verbs, the type 19 people are appointed and replaced; the type 29 people are given orders and permissions.[6]

Several types are dominated by a common and semantically empty verb, such as *be* (12), *have* (23), or *make* (31). In these cases, the noun sets are not intuitively coherent, presumably because these verbs impose such weak selectional restrictions. The verb sets are not particularly coherent either, though this is balanced by the fact that most of the verbs have low probabilities. These common verbs also occur in many other types, for instace *be* in top position in type 32, which is an intuitively coherent type.

[6]In verb group 19, a number of items resulting from parsing errors are evident. Presumably, *manage* comes from *managing director* misidentified as a verb phrase.

type 1 .015				type 2 .021			
meet	.23606	standard	.06597	end	.10515	year	.11873
set	.15089	record	.05207	trade	.07562	Friday	.09733
be	.04807	goal	.04920	say	.06728	week	.08977
keep	.04205	need	.04453	work	.06685	month	.08074
hit	.03637	requirement	.04152	begin	.06120	day	.06488
miss	.03403	level	.03564	spend	.05976	time	.04695
exceed	.03322	demand	.03168	announce	.02880	today	.04067
reach	.02533	target	.02963	close	.02638	Tuesday	.03573
achieve	.01923	high	.02828	open	.02420	hour	.02881
fulfill	.01518	stage	.02675	start	.02219	Monday	.02724
satisfy	.01330	pace	.02155	expire	.01716	way	.02458
live	.01241	date	.01860	mark	.01618	Wednesday	.02287
surpass	.01163	payment	.01819	last	.01545	capital	.02116
establish	.01126	expectation	.01764	wait	.01396	talk	.01933
stress	.00919	point	.01623	follow	.01147	night	.01526
eliminate	.00858	limit	.01598	serve	.01117	war	.01030

type 3 .032				type 4 .029				type 5 .035			
use	.15863	product	.07639	continue	.06015	operation	.08075	complete	.08198	sale	.10672
develop	.09296	system	.04459	begin	.05647	effort	.07703	finance	.05656	acquisition	.06132
produce	.08613	drug	.03406	launch	.04128	program	.06099	include	.04693	transaction	.05817
introduce	.03273	technology	.03110	conduct	.03157	campaign	.02961	be	.03700	change	.04977
market	.02933	computer	.02236	resume	.02845	production	.02943	say	.03244	purchase	.04142
sell	.02778	country	.01948	expand	.02528	investigation	.02209	approve	.03176	merger	.03092
include	.02140	car	.01875	start	.02476	service	.02029	represent	.02703	order	.02709
ship	.02109	equipment	.01685	be	.02280	process	.01670	announce	.02536	increase	.02646
supply	.01770	line	.01666	say	.02159	work	.01508	block	.02272	action	.02278
get	.01602	version	.01500	halt	.01456	negotiation	.01414	consider	.02208	takeover	.01939
distribute	.01149	machine	.01410	follow	.01335	payment	.01407	involve	.01758	project	.01827
test	.01145	proceed	.01398	ban	.01304	activity	.01383	propose	.01702	issue	.01792
manufacture	.01107	program	.01230	restrict	.01189	practice	.01290	explore	.01670	offering	.01475
promote	.01035	ton	.01226	support	.01187	talk	.01264	expect	.01570	move	.01278
install	.00983	software	.01178	oversee	.01148	development	.01256	follow	.01496	use	.01205
build	.00971	model	.01146	plan	.01145	use	.01224	discuss	.01474	investment	.01098

type 6 .024				type 7 .023				type 8 .041			
file	.14736	suit	.11129	play	.09036	money	.20670	raise	.17768	price	.15348
follow	.09798	case	.06304	spend	.07740	$	.07810	increase	.09248	rate	.15152
settle	.04932	report	.06176	raise	.05364	role	.05575	boost	.07753	stake	.04734
deny	.03748	decision	.04991	be	.03725	fund	.05401	reduce	.04806	question	.03490
issue	.03688	charge	.04782	get	.03375	cash	.03552	lower	.04484	rating	.03048
hear	.03011	lawsuit	.03991	lose	.03372	lot	.02660	cut	.03509	value	.02881
say	.02876	statement	.03674	cost	.03356	time	.02655	say	.02377	sale	.02544
dismiss	.02466	appeal	.02439	save	.03059	dollar	.02612	offer	.02240	earning	.02379
bring	.02329	complaint	.02390	use	.02727	part	.02037	push	.01979	number	.02363
be	.02301	claim	.02312	put	.02145	million	.01805	expect	.01605	level	.02135
confirm	.02157	allegation	.02142	lend	.01760	game	.01680	keep	.01407	dollar	.01719
appeal	.02020	ruling	.01916	pay	.01726	capital	.01508	disclose	.01198	capital	.01707
include	.02019	plan	.01561	invest	.01698	total	.01425	double	.01191	revenue	.01661
review	.01744	action	.01490	need	.01668	billion	.01380	answer	.01182	size	.01569
reverse	.01527	rumor	.01405	give	.01620	life	.01291	bring	.01062	cost	.01369
see	.01521	recommendation	.01402	add	.01526	year	.01152	maintain	.01042	production	.01367

type 9 .023				type 10 .014				type 11 .032			
reject	.08033	bid	.15795	hold	.42625	company	.29762	pay	.21425	cost	.13843
consider	.06539	proposal	.11781	acquire	.04866	meeting	.09623	reduce	.14585	debt	.08962
accept	.05285	offer	.11493	call	.04400	stake	.05578	cut	.09985	tax	.05801
support	.03535	plan	.06008	attend	.02986	talk	.04176	increase	.03444	dividend	.03704
decline	.03064	comment	.03193	schedule	.02436	hearing	.03762	cover	.02748	price	.03451
drop	.02975	request	.02591	leave	.02433	conference	.03723	include	.02187	$	.02684
close	.02525	idea	.02331	say	.02301	position	.02211	be	.01433	fee	.02370
submit	.02401	attempt	.01774	tell	.02173	election	.01934	repay	.01355	expense	.02319
be	.02389	claim	.01346	force	.02111	concern	.01578	slash	.01313	deficit	.02154
receive	.02352	effort	.01319	follow	.01705	discussion	.01367	keep	.01256	interest	.01911
approve	.02158	issue	.01301	be	.01290	post	.01207	limit	.01225	bill	.01868
launch	.02135	strategy	.01216	mine	.01207	stock	.01060	control	.01161	loan	.01818
back	.01941	option	.01175	control	.01057	hostage	.01017	avoid	.00944	capital-gains	.01785
withdraw	.01771	application	.01093	seek	.00993	party	.00770	trim	.00897	premium	.01715
review	.01711	amendment	.01044	expect	.00783	session	.00729	raise	.00868	amount	.01678
announce	.01688	$	.00988	value	.00756	interest	.00715	collect	.00847	force	.01579

type 12 .093				type 13 .025				type 14 .030			
be	.86747	part	.03984	do	.20440	business	.15112	show	.10237	interest	.07597
become	.02744	way	.02670	run	.10726	job	.09334	reflect	.05022	growth	.05279
find	.01918	time	.02601	be	.09109	thing	.05915	express	.03955	value	.03758
see	.00639	president	.02546	create	.06320	company	.03691	see	.03130	economy	.03269
give	.00600	company	.01500	form	.05178	work	.03633	improve	.02787	concern	.03047
include	.00383	reason	.01314	leave	.03496	venture	.02406	slow	.01795	demand	.03012
get	.00316	lot	.01245	get	.03488	lot	.01903	say	.01794	sign	.02552
identify	.00295	problem	.01181	find	.01891	government	.01850	enhance	.01611	performance	.02153
remain	.00257	chairman	.01181	start	.01579	fund	.01547	continue	.01535	return	.02080
cite	.00242	issue	.01158	see	.01489	program	.01358	pursue	.01512	confidence	.02062
represent	.00222	case	.01024	eliminate	.01072	deal	.01143	grow	.01505	ability	.02002
seem	.00217	unit	.00933	keep	.00987	ad	.01139	fuel	.01503	strength	.01637
prove	.00217	sign	.00926	lose	.00979	risk	.00982	indicate	.01460	inflation	.01435
allow	.00201	thing	.00897	expect	.00978	year	.00949	cite	.01430	benefit	.01353
want	.00200	target	.00818	enter	.00822	system	.00826	represent	.01363	support	.01348
mark	.00197	question	.00776	establish	.00798	room	.00784	expect	.01304	improvement	.01302

type 15 .022				type 16 .016			
reach	.14872	agreement	.23792	send	.13358	letter	.06276
sign	.11051	plan	.15452	carry	.08868	dividend	.05273
announce	.08203	bill	.08091	receive	.06249	warrant	.04434
approve	.06145	contract	.04619	write	.05310	message	.03976
pass	.04008	legislation	.03891	declare	.04667	book	.02798
be	.03342	letter	.02473	get	.04335	note	.02597
negotiate	.02940	accord	.02395	publish	.03554	signal	.02243
say	.02620	settlement	.02249	issue	.02821	stock	.02046
introduce	.01946	deal	.01928	read	.02800	article	.02037
terminate	.01562	measure	.01622	deliver	.02297	information	.01688
spend	.01310	pact	.01546	include	.01517	sentence	.01583
veto	.01293	package	.01450	serve	.01473	ad	.01531
adopt	.01238	level	.01253	subordinate	.01435	price	.01447
propose	.01171	resolution	.01110	see	.01412	news	.01350
forge	.01142	program	.01028	suspend	.01406	report	.01294
discuss	.01009	decision	.00951	regard	.01123	copy	.01098

Table 6: Half of a 32 type selection model.

type 17 .020			
increase	.08773	market	.25364
expand	.07881	pressure	.06255
put	.06053	business	.03365
enter	.05256	board	.02548
be	.03619	capacity	.02433
grow	.02317	industry	.02232
tap	.01821	membership	.02207
dominate	.01791	line	.01559
keep	.01595	economy	.01522
open	.01550	area	.01442
serve	.01435	base	.01225
hit	.01415	end	.01221
putt	.01408	competition	.01185
affect	.01397	country	.01128
bring	.01250	sale	.01123
help	.01201	head	.01119

type 18 .021			
violate	.07151	law	.08733
impose	.04621	right	.08309
adopt	.04096	rule	.07114
ease	.03129	policy	.05715
exercise	.03047	provision	.04352
include	.02975	restriction	.04050
tighten	.02341	credit	.03410
propose	.02050	ban	.03290
enforce	.01916	regulation	.02586
use	.01698	security	.02519
change	.01655	control	.02277
break	.01621	option	.02191
extend	.01587	standard	.01958
lift	.01441	requirement	.01490
oppose	.01376	tax	.01227
remove	.01323	duty	.01227

type 19 .031			
say	.26679	director	.08356
become	.11146	president	.06178
manage	.07970	official	.05558
be	.06654	analyst	.04518
remain	.05788	chairman	.04354
tell	.03499	executive	.04324
include	.03194	manager	.03607
name	.02965	partner	.02709
hire	.02161	member	.02248
market	.01336	trader	.01837
act	.00933	board	.01599
elect	.00929	firm	.01408
oust	.00924	lawyer	.01362
ask	.00854	company	.01347
appoint	.00786	consultant	.01221
replace	.00770	banker	.01196

type 20 .015			
assume	.05587	position	.12880
fill	.04523	account	.06505
retain	.04154	control	.04908
return	.03565	power	.04743
be	.02827	image	.04092
share	.02387	responsibility	.03439
strengthen	.02213	post	.03189
maintain	.02148	call	.03184
seize	.02047	view	.01745
bear	.01857	home	.01714
change	.01815	ownership	.01703
improve	.01648	job	.01595
use	.01636	order	.01501
shift	.01626	name	.01296
place	.01532	title	.01296
bolster	.01465	seat	.01233

type 21 .016			
join	.12423	firm	.13008
lead	.09514	mortgage	.06931
cap	.07303	group	.05734
bank	.07174	company	.05733
consult	.06777	attention	.05347
head	.05825	concern	.03268
draw	.04459	force	.02934
indicate	.03417	coupon	.02364
engineer	.02217	list	.02076
trade	.01944	indicator	.01819
attract	.01666	unit	.01787
form	.01528	office	.01772
stage	.01513	board	.01306
focus	.01419	operation	.01214
turn	.01066	team	.01078
create	.01065	way	.01048

type 22 .025			
operate	.23595	officer	.08099
build	.12124	plant	.08029
open	.05096	profit	.05795
close	.03310	system	.04020
manufacture	.02973	store	.03467
keep	.02182	facility	.03319
run	.01790	operation	.02793
turn	.01747	office	.02522
establish	.01689	center	.02376
move	.01551	car	.01748
be	.01425	company	.01672
fly	.01396	home	.01634
expand	.01335	door	.01471
say	.01300	mile	.01144
maintain	.01149	building	.01131
lease	.01147	eye	.01127

type 23 .053			
have	.88786	loss	.03589
be	.02008	share	.03026
say	.00934	effect	.02858
get	.00734	impact	.02606
lose	.00644	sale	.02217
see	.00606	interest	.02082
limit	.00577	income	.01999
lack	.00458	problem	.01842
increase	.00432	right	.01549
cite	.00375	plan	.01473
cast	.00366	time	.01334
add	.00228	chance	.01282
exceed	.00220	trouble	.01257
feel	.00208	comment	.01108
consider	.00198	value	.01099
mean	.00194	asset	.01065

type 24 .058			
sell	.24706	share	.17737
buy	.21141	stock	.09721
acquire	.06323	stake	.06503
purchase	.04274	company	.03334
own	.03560	interest	.03095
include	.03528	asset	.03044
total	.02347	unit	.02787
hold	.02227	bond	.02452
be	.01894	business	.02242
issue	.01459	security	.01817
prefer	.01360	operation	.01475
trade	.01205	dollar	.01390
retain	.00957	ton	.01164
offer	.00840	issue	.01091
receive	.00702	product	.00996
increase	.00684	car	.00955

type 25 .031			
receive	.15526	approval	.09046
get	.14544	contract	.08712
win	.11622	control	.04008
seek	.11268	order	.02775
gain	.07195	support	.02298
obtain	.03624	share	.01816
give	.02811	damage	.01665
require	.02492	license	.01496
lose	.02422	access	.01461
need	.02124	help	.01358
loose	.01948	benefit	.01258
grant	.01426	vote	.01237
demand	.01242	right	.01227
secure	.01030	loan	.01140
award	.00975	attention	.01126
await	.00788	boost	.01075

type 26 .054			
rise	.22903	%	.74167
yield	.13168	point	.07754
fall	.12219	cent	.05860
be	.08785	yen	.01537
own	.03445	pence	.01498
drop	.02992	price	.00969
jump	.02585	rate	.00935
grow	.02474	year	.00776
increase	.02322	average	.00451
climb	.02222	cost	.00328
decline	.02091	bit	.00276
gain	.02001	demand	.00253
hold	.01557	ton	.00250
buy	.01167	fee	.00244
acquire	.01089	inflation	.00244
soar	.00896	range	.00231

type 27 .029			
provide	.22624	service	.05067
offer	.11291	way	.04967
change	.08438	information	.04529
give	.06825	detail	.04282
find	.04567	hand	.04176
disclose	.03263	name	.02648
get	.03251	data	.02517
be	.01960	benefit	.01551
seek	.01937	term	.01436
discuss	.01320	incentive	.01346
use	.01279	figure	.01076
clear	.01147	evidence	.01069
include	.01115	$	.01045
obtain	.01086	loan	.01006
need	.01085	reason	.00998
release	.00901	record	.00987

type 28 .025			
take	.71341	yesterday	.08897
trade	.21074	place	.06415
handle	.00380	advantage	.05361
offer	.00343	step	.04657
call	.00334	action	.03615
include	.00324	effect	.03290
see	.00264	volume	.03119
find	.00205	charge	.02457
offset	.00204	control	.02322
represent	.00179	position	.01873
enjoy	.00148	year	.01862
remain	.00146	profit	.01806
accept	.00121	look	.01770
give	.00120	time	.01435
overcome	.00117	risk	.01336
note	.00112	part	.01320

type 29 .052			
allow	.05595	company	.07287
give	.04996	people	.05791
tell	.04769	investor	.04724
help	.04348	government	.02802
ask	.03068	customer	.02600
require	.02369	employee	.02555
say	.02168	worker	.02435
attract	.02134	client	.01987
force	.02101	bank	.01928
represent	.02067	shareholder	.01402
protect	.01823	agency	.01363
leave	.01805	consumer	.01271
urge	.01768	group	.01270
keep	.01565	state	.01254
include	.01482	reporter	.01234
encourage	.01475	court	.01153

type 30 .037			
report	.19567	loss	.17676
post	.12245	gain	.10309
say	.07210	earning	.09973
expect	.05355	profit	.08199
include	.04197	income	.05284
operate	.02780	sale	.04715
show	.02579	decline	.04575
attribute	.02379	increase	.04519
follow	.02049	result	.04247
have	.01666	rise	.02973
generate	.01453	drop	.02688
produce	.01249	revenue	.02411
be	.01199	event	.01580
see	.01023	net	.01198
reflect	.01005	return	.01168
estimate	.00984	charge	.00803

type 31 .029			
make	.83180	decision	.05421
say	.01986	money	.03619
call	.00735	payment	.03542
see	.00696	sense	.03247
expect	.00582	bid	.02889
accept	.00552	offer	.02289
welcome	.00513	product	.02105
use	.00451	change	.02097
avoid	.00400	loan	.01927
leave	.00360	move	.01693
keep	.00357	profit	.01682
affect	.00315	investment	.01484
guarantee	.00299	difference	.01479
cancel	.00278	effort	.01280
defer	.00277	statement	.01278
reflect	.00250	$	.01184

type 32 .029			
be	.11257	problem	.14799
face	.08331	issue	.02183
cause	.04889	pressure	.02099
resolve	.02475	damage	.02080
solve	.02389	recession	.01835
avoid	.02373	challenge	.01664
address	.02302	situation	.01663
fight	.02070	effect	.01617
create	.02057	competition	.01527
pose	.01971	threat	.01394
reflect	.01856	risk	.01369
see	.01673	concern	.01343
prevent	.01505	question	.01327
ease	.01416	crisis	.01323
suffer	.01239	shortage	.01284
cite	.01236	crime	.01239

Table 7: The other half.

5 Application to parsing

A casual examination of clusters can at most suggest that the right sort of thing is going on; the point is to use such representations to do something that we want to do for an independent reason. In the introduction, I said that resolving many parsing ambiguities comes down to evaluating selectional compatibility between two lexical items. Given a probabilistic selection model, we can assign a probability to any verb-object pair drawn from the lexicon of 992 verbs and 1027 nouns. In order to evaluate parses, we need to include other grammatical relations. In some cases, such as subjects, this is fairly straightforward. In others, such as second complements of verbs, getting access to frequency counts is not straightforward, since identifying second complements — such as prepositional phrases — requires resolving attachment ambiguities. This can not be done systematically without the kind of lexical information we are trying to induce. Presumably, a procedure learning selectional restrictions for second complements would have to initially consider several attachments, and iteratively learn the lexical information required for disambiguation. This method is applied to simpler data (paying attention just to prepositions and not the heads of their objects) in Hindle and Rooth [Hindle and Rooth, 1993].

To investigate the possibility of disambiguating syntactic ambiguities with selectional information, I considered the past participle vs. tensed verb ambiguity of *sold* in positions immediately following a noun phrase. In the first example below, *sold* is a tensed verb, and the preceding noun *administration* is the head of its subject, describing the agent. In the second example, *sold* is a participial post-modifier, and the preceding noun phase denotes the sold object.

(11)a. 10604815 evidence that the Reagan administration sold arms to Iran
b. 3001228 represented payments for arms sold to Nicaraguan insurgents

The situation is actually more complex, since *sold* occurs frequently in the middle construction, with a syntactic subject filling the semantic role of a sold object, rather than the seller:

(12) 2979922 The franchise sold in 1979 for $11 million

To sidestep this problem, I defined the problem to be solved as one of identifying the semantic relation (seller or sold object) of the noun phrase, rather than identifying a grammatical relation or part of speech. In other words, the middle constructions are grouped with the postmodifiers. By hand, I classified the first relevant occurrence of each noun-*sold* pair in the full Wall Street Journal corpus from [Liberman, 1992]. Of the 1027 nouns in the model, 220 were represented in this configuration, of which 87 were past tense verbs, 118 were participial postmodifiers, and 15 middle constructions. An augmented selection model was obtained (in a somewhat ad hoc way) by adding two ad-

ditional verbs sold/VBN and sold/VBD to the verb set. Training material consisted similar but independent data.[7] The model was used to classify the 220 test examples by means of the ratio:

$$\frac{p_{\text{sold/VBD},n}}{p_{\text{sold/VBD},n} + p_{\text{sold/VBN},n}}$$

Pairs with a score greater than 0.5 were assigned to the seller role, and those with a lower score to the sold object role. Of the 118 post-modifier pairs, 110 were correctly classified into the sold object role. Of the 87 ordinary subject-verb items, 69 were correctly classified into the seller role. Of the 15 instances of the middle construction, 11 were correctly classified into the sold object role. This gives an rate of correct classification of 86.4%. The disambiguation scores are listed in tables 5, 5, and 5. Appositely, the least ambiguous instance of the seller role is the noun *seller*. The least ambiguous example of the sold object role is *output*. Many of the problematic nouns — those with scores below 0.5 in table 5 — name institutions which can be agents, but can also be bought and sold, for instance *company*, *airline*, and *store*. Since companies are actually described in the Wall Street Journal both as being sold and as selling things, we would not expect a selectional approach to be uniformly successful in this case. However, we want to resolve clear cases correctly — artefacts, materials, and financial instruments are unambiguos sold objects, and people are unambiguous sellers. With few exceptions, such clear cases are resolved correctly.

6 Matrix formulation

The definition of $p_{v,n}$ from section 3 can be written as a matrix product. Let L be the $V \times k$ matrix representing the verb probabilities, $L_{i,\tau} = p_i^\tau$, let R be the $N \times k$ matrix representing the noun probabilities, $R_{j,\tau} = p_j^\tau$, and let D be a diagonal matrix representing the type probabilities, $D_{\tau,\tau} = p_\tau$. Then a derived probability distribution on $V \times N$ is given by a matrix product:

$$LDR^T \qquad [L_{v/t} D_{t/t} [R^T_{n/t}]_{t/n}]_{v/n} \tag{1}$$

In the version on right, the subscripts give matrix dimensions in categorial notation, v being the verb cardinality, t the type cardinality, and n the noun cardinality; R^T is the transpose of the matrix R, $[R^T]_{i,j} = R_{j,i}$.

Proposition. Suppose L,D, and R are probability matrices as described above, that is:

$$\forall\tau \left[\textstyle\sum_i L_{i,\tau} = 1\right] \quad \forall\tau \left[\textstyle\sum_j R_{j,\tau} = 1\right] \quad \textstyle\sum_\tau D_{\tau,\tau} = 1$$

[7] In writing this section, I found that the training data and my record of how they were constructed had been lost. Therefore, the evaluation will be redone, and final results may differ from those described here.

output	.00000	volume	.00001	suit	.01225	gasoline	.04458
bill	.04526	missile	.05726	movie	.07933	system	.08979
technology	.09182	package	.09475	film	.09783	advertising	.11294
material	.11427	shoe	.11488	oil	.12449	product	.12479
chip	.12479	plant	.12868	device	.13161	merchandise	.13183
good	.13240	car	.13363	machine	.13475	operation	.13514
barrel	.13680	building	.13923	loan	.14066	document	.14141
gold	.14160	copy	.14185	debt	.14402	truck	.14454
debenture	.14468	bond	.14542	brand	.14633	share	.14641
gas	.14673	food	.14708	coverage	.14758	certificate	.14839
inventory	.14886	stake	.14924	asset	.14957	acre	.15149
phone	.15157	property	.15334	note	.15414	version	.15509
warrant	.15515	ticket	.15528	block	.15751	card	.15806
pound	.16148	business	.16485	home	.16962	contract	.17100
computer	.17181	steel	.17360	show	.17449	amount	.17630
dollar	.18039	program	.18111	aircraft	.18204	insurance	.18227
engine	.18332	book	.18543	site	.19478	line	.19487
land	.19565	drug	.19588	security	.19686	weapon	.19701
test	.19931	tape	.20481	item	.21203	house	.21365
rest	.21706	plane	.21825	station	.21854	piece	.22123
paper	.22272	mark	.22307	import	.22753	arm	.23139
model	.23296	magazine	.23470	call	.24358	billion	.24488
service	.25488	issue	.25688	work	.25714	policy	.26237
energy	.26286	supply	.26545	position	.29452	vehicle	.29797
thrift	.30081	collection	.30386	list	.33147	ad	.33409
million	.33923	game	.34198	one	.34246	shipment	.38118
art	.38336	article	.39738	newspaper	.39756	type	.41898
set	.43282	transaction	.45646				
hospital	.55241	kind	.68506	player	.69998	export	.71760
switch	.74617	agent	.75588	animal	.84308	premium	.84474

Table 8: Disambiguation scores for NP/postmodifier combinations. Items above the line are correctly classified.

trade	.00001	effort	.08683	plan	.11405	shop	.14468
estate	.16620	division	.17522	store	.17609	unit	.23540
subsidiary	.26404	account	.27786	network	.32662	fund	.32709
company	.36211	trust	.39613	partnership	.40965	concern	.43930
airline	.44054	institution	.46481				
giant	.50438	team	.54480	parent	.59502	organization	.63279
operator	.63752	utility	.63796	insurer	.64974	bank	.66487
industry	.66728	couple	.67811	manager	.68160	maker	.70603
country	.71725	firm	.72093	manufacturer	.72645	shareholder	.74399
foreigner	.75349	room	.75649	lender	.75983	group	.77943
corporation	.78073	husband	.80379	publisher	.80626	developer	.81175
nation	.83127	stockholder	.83832	state	.87692	family	.88925
school	.89850	broker	.90178	individual	.90353	executive	.90404
producer	.90640	agency	.91013	partner	.93265	customer	.94037
wife	.94645	chairman	.94838	holder	.95723	board	.95960
man	.96147	department	.96479	father	.96875	dealer	.97063
people	.97064	member	.97166	investor	.97463	brother	.97580
employee	.97603	plaintiff	.97753	world	.97878	government	.98305
defendant	.98917	farmer	.99021	friend	.99417	lawyer	.99551
trader	.99737	official	.99850	client	.99883	owner	1.00000
administration	1.00000	analyst	1.00000	buyer	1.00000	director	1.00000
foundation	1.00000	officer	1.00000	participant	1.00000	president	1.00000
seller	1.00000						

Table 9: Disambiguation scores for agent/verb combinations. Items below the line are correctly classified.

offering	.13354	metal	.14244	apartment	.14495	stock	.14550
franchise	.15312	membership	.16785	future	.19601	seat	.27562
price	.43852	hundred	.44699	conference	.49000		
other	.57470	thing	.65876	target	.99409	run	.99997

Table 10: Disambiguation scores for middle constructions. Items above the line are correctly classified.

Then $\sum_{i,j} [LDR^T]_{i,j} = 1$. In the verification below, $M_{i.}$ and $M_{.j}$ denote the ith row and jth column of a matrix M, respectively.

$$\begin{array}{rcll}
[DR^T]_{\tau,j} & = & D_{\tau.} \cdot [R^T]_{.j} & \\
 & = & D_{\tau.} \cdot R_{j.} & \\
 & = & D_{\tau,\tau} R_{j,\tau} & \text{(since } D \text{ is diagonal)} \\
[LDR^T]_{i,j} & = & L_{i.} \cdot [DR^T]_{.j} & \\
 & = & \sum_\tau L_{i,\tau} [DR^T]_{\tau,j} & \\
 & = & \sum_\tau L_{i,\tau} D_{\tau,\tau} R_{j,\tau} & \text{(previous equality)} \\
\sum_{i,j} [LDR^T]_{i,j} & = & \sum_{i,j} \sum_\tau L_{i,\tau} D_{\tau,\tau} R_{j,\tau} & \text{(previous equality)} \\
 & = & \sum_\tau D_{\tau,\tau} \left(\sum_i L_{i,\tau} \left(\sum_j R_{j,\tau} \right) \right) & \\
 & = & \sum_\tau D_{\tau,\tau} \left(\sum_i L_{i,\tau} \right) & \text{(second assumption)} \\
 & = & \sum_\tau D_{\tau,\tau} & \text{(first assumption)} \\
 & = & 1 & \text{(third assumption)}
\end{array}$$

So, we are justified in describing the matrix product as a probability distribution on $V \times N$. This representation is reminiscent of singular value decompositon of matrices (SVD): in fact the such a decompositon takes exactly the form (1). However, the constraints on L and R are different: in SVD, they are orthonormal, meaning that any two distinct rows a null dot product, and the dot product on any row with itself is 1. These conditions are different from those imposed by the interpretation as probability matrices. One symptom of this is that a SVD approximation of a frequency matrix may contain negative entries.

A further difference has to do with the relation beween the product matrix and the original data. In SVD, the product provides the best least squares fit to the frequency matrix, of a given rank. What is being optimized in the probability model is most easily understood as the probability of the observed sequence of verb-noun pairs. This can be written:

$$\prod_{v,n} p_{v,n}^{f_{v,n}}$$

In entropy terms, we seek to minimize

$$\sum_{v,n} f_{v,n} \left(-\log_2 p_{v,n} \right)$$

This is quite different from the least-squares criterion.

References

[Church et al., 199x] Church, K. W., Gale, W. A., Hanks, P., and Hindle, D. (199x). Using statistics in lexical analysis. In Zernik, editor, *Lexical acquisition: exploiting on-line resources to build a lexicon.*

[Hindle, 1983] Hindle, D. (1983). User manual for fidditch, a deterministic parser. Technical Memorandum 7590-142, Naval Research Laboratory.

[Hindle, 1993] Hindle, D. (1993). A parser for text corpora. In Atkins, B. and Zampolli, A., editors, *Computational Approaches to the Lexicon.* Oxford University Press, Oxford.

[Hindle and Rooth, 1993] Hindle, D. and Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, 18:xx–yy.

[Liberman, 1992] Liberman, M. (1992). *ACL-DCI CD ROM 1.* Association for Computational Linguistics.