Prediction of Lexicalized Tree Fragments in Text

Donald Hindle

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

ABSTRACT

There is a mismatch between the distribution of information in text, and a variety of grammatical formalisms for describing it, including ngrams, context-free grammars, and dependency grammars. Rather than adding probabilities to existing grammars, it is proposed to collect the distributions of flexibly sized partial trees. These can be used to enhance an ngram model, and in analogical parsing.

1. THE PROBLEM WITH PROBABILIZED GRAMMARS

For a variety of language processing tasks, it is useful to have a predictive language model, a fact which has recently led to the development probabilistic versions of diverse grammars, including ngram models, context free grammars, various dependency grammars, and lexicalized tree grammars. These enterprises share a common problem: there is a mismatch between the distribution of information in text and the grammar model.

The problem arises because each grammar formalism is natural for the expression of only some linguistic relationships, but predictive relationships in text are not so restricted. For example, context-free grammars naturally express relations among sisters in a tree, but are less natural for expressing relations between elements deeper the tree. In this paper, first we discuss the distribution of information in text, and its relationship to various grammars. Then we show how a more flexible grammatical description of text can be extracted from a corpus, and how such description can enhance a language model.

Ngram Models The problem can be seen most simply in ngram models, where the basic operation is to guess the probability of a word given $n - 1$ previous words. Obviously, there is a deeper structure in text than an n-gram model admits, though thus far, efforts to exploit this information have been only marginally successful. Yet even on its own terms, ngram models typically fail to take into account predictive information.

One way that ngram models ignore predictive information is in their strategy for backing off. Consider, for example, a trigram model where the basic function is to predict a word (w_0) given the two previous words (w_{-1} and w_{-2}). In our Wall Street Journal test corpus, the three word sequence *give kittens to* appears once, but not at all in the training corpus. Thus, a trigram model will have have difficulty predicting *to* given the words *give kittens*.

In this case, the standard move of backing off to a bigram model is not very informative. It is more useful to predict *to* using the word *give* than the word *kittens*, because we know little about what can follow *kittens*, but much about what typically follows *give*. We would expect for cases where the bigram (w_{-1},w_0) does not exist, the alternative bigram (w_{-2},w_0) will be a better predictor (if it exists) than the simple unigram.

Obviously, in this example, the fact that complementation in English is not expressed purely by adjacency explains some of the power of the w_{-1} predictor.

A second problem with ngram models arises because different word sequences call for a greater or smaller n. For example, while many 6-grams are unique and uninformative, some are powerful predictors.

Table 1 shows the frequencies of the top few words following the words *New York Stock Exchange* in the 60 million word Wall Street Journal corpus. More than half the time, the word that follows *New York Stock Exchange* is *composite*. However, in the 355 cases where *New York Stock Exchange* is preceded by the word *composite* (Table 1), *composite* never occurs as the following word, and the overwhelming probable choice for the following word is *trading*.

If we had settled for a 5-gram model here, we would have failed miserably compared with a 6-gram model. But of course, this raises the sparse data problem; predicting the parameters of a 6-gram model is daunting.

Context Free Grammars It is easy to see that a simple-minded probabilizing of a CFG – that is, taking an existing CFG and assigning probabilities to the rules – is not a very good predictor. There several problems. First, CFG's typically don't include enough lexical information. Indeed, the natural use of non-terminal categories is to abstract away from

New York Stock Exchange	*composite*	6597
	,	1556
	yesterday	862
	.	824
	trading	480
	..	
	TOTAL	12305
composite New York Stock Exchange	*trading*	349
	yesterday	4
	Trading	2
	composite	0
	..	
	TOTAL	355

Table 1: Ngrams with *New York Stock Exchange*

lexical considerations. Lexical associations are however critical to guessing word probabilities, not only for verb subcategorization and selection, but across the vocabulary (see e.g. Church et al. 1991). A context free grammar with a rule $N2- > ADJ * N$ is not able to naturally express selectional restrictions between adjectives and nouns, e.g. the fact that *strong tea* is probable but *powerful tea* is not.

A second problem is that CFG's naturally abstract away from syntactic function: for example, in a CFG, a noun phrase is described by the same set of rules whether it occurs as subject, object, object of preposition or whatever. While this ability to generalize across contexts is a strength of CFG's, it is disastrous for guessing whether a noun phrase will be a pronoun or not. Table 2 shows the probabilities of a noun phrase being realized as a pronoun in various contexts, in a sample of spoken and written texts produced by college students and matched for content (Hindle 1978). Clearly, ignoring whether a noun phrase is subject or not reduces the effectiveness of a predictive model. (Note too that the differences between spoken and written English are not to be ignored.

There are of course ways to admit lexical and functional information into a CFG. But except for carefully restricted domains (e.g semantic grammars), these typically lead to an explosion of nonterminals and rules, making parameter estimation difficult.

	function	p(PRO)
spoken	subject	.71 (N=2077)
	non-subject	.16 (N=1477)
written	subject	.44 (N=1195)
	non-subject	.09 (N=1088)

Table 2: Subject and non-subject noun phrases

Dependency Grammars Dependency grammars naturally address part of the mismatch between CFG's and predictive associations, since they are expressed in terms of relations between words (Melcuk 1988). Nevertheless, in dependency grammars as well, certain syntactic relationships are problematic.

In dependency grammar, there are two competing analyses both for noun phrases and for verb phrases. For noun phrases, the head may be taken to be either 1) the head noun (e.g. *man* in *the men*) or 2) the determiner (e.g *the* in *the men*); analogously, for verb phrases, the head may be taken to be either 1) the mail verb (e.g. *see* in *had seen*) or 2) the tensed verb of the verb group (e.g *have* in *had seen*). Each analysis has its virtues, and different dependency theorists have preferred one analysis or the other. It is not our purpose here to choose a dependency analysis, but to point out that whatever the choice, there are consequences for our predictive language models. The two models imply different natural generalizations for estimating probabilities, and thus will lead to different predictions about the language probabilities. If the determiner is taken to be the head of the noun phrase, then in guessing the probability of a *verb-det-noun* structure, the association between the verb and the determiner will predominate, since when we don't have enough information about a *verb-det-noun* triple, we can back off to pairs. Conversely, if the noun is taken to be the head of the noun phrase, then the predominant association will be between verb and noun. (Of course, a more complex relationship between the grammar and the associated predictive language model may be defined, overriding the natural interpretation.)

A ten million word sample of *Wall Street Journal* text was parsed, and a set of *verb-det-noun* triples extracted. Specifically, object noun phrases consisting of a noun preceded by a single determiner preceded by a verb were tabulated. That is, we consider only verbs with an object, where the object consists of a determiner and a noun. The five most common such triples (preceded by their counts) were:

213	*have*	*a*	*loss*
176	*be*		
165	*be*	*the*	*first*
140	*raise*	*its*	*stake*
127	*reach*	*an*	*agreement*

Three different probability models for predicting the specific verb, determiner, and noun were investigated, and their entropies calculated. Model 0 is the baseline trigram model, assuming no independence among the three terms. Model 1, the natural model for the determiner=head dependency model, predicts the determiner from the verb and the noun from the determiner (and thus is equivalent to an adjacent word bigram model). Model 2 is the converse, the natural model for the noun=head dependency model. Both Model 1 and Model 2

	Model for $[_{VP}\ v\ [_{N}P\ d\ n\ ]]$	Entropy
0	$Pr(vdn) = Pr(v)Pr(dn\|v)$	15.08
1	$Pr(vdn) = Pr(v)Pr(d\|v)Pr(n\|d)$	20.48
2	$Pr(vdn) = Pr(v)Pr(n\|v)Pr(d\|n)$	17.62

Table 3: Three predictive models for verb-det-noun triples in Wall Street Journal text

ignore predictive information, assuming in the first case that the choice of noun is independent of the verb, and in the second case, that the choice of determiner is independent of the verb. Neither assumption is warranted, as Table 3 shows (both have higher entropy than the trigram model), but Model 1, the determiner=head model, is considerably inferior. Model 1 is for this case like a bigram model, and Table 3 makes it clear that this is not a particularly good way to model dependencies between verb and object: the dominant dependency is between verb and noun.

In terms of using the distributional information available in text, neither choice is correct, since the answer is lexically specific. For example, in predicting the object of verbs, *answer* is a better predictor of its object noun (*call, question*), while *alter* is better a predicting its determiner (*the, its*).

In contrast to dependency grammars and context free grammars, lexicalized tree adjoining grammars have considerable flexibility in what relations are represented, since the tree is an arbitrary-sized unit (Shabes 1988). In practice however, lexicalized TAGs have typically been written to reduce the number of rules, and thus to assume independence like other grammars. In general, for any grammar that is written without regard to the distribution of forms in text, simply attaching probabilities to the grammar will always ignore useful information. This does not imply any claim about the descriptive power of various grammar formalisms; with sufficient ingenuity, just about any recurrent relation that appears in a corpus can be encoded in any formalism. However, different grammar formalisms do differ in what they can *naturally* express.

There is a clear linguistic reason for the mismatch between received grammars and the distribution of structures in text: language provides several cross cutting ways of organizing information (including various kinds of dependencies, parallel structures, listing, name-making templates, etc.), and no single model is good for all of these.

2. USING PARTIAL STRUCTURES

The preceding section has given evidence that adding probabilities to existing grammars in several formalisms is less than optimal since significant predictive relationships are necessarily ignored. The obvious solution is to enrich the grammars to include more information. To do this, we need variable sized units in our database, with varying terms of description, including adjacency relationships and dependency relationships. That is, given the unpredictable distribution of information in text, we would like to have a more flexible approach to representing the recurrent relations in a corpus. To address this need, we have been collecting a database of partial structures extracted from the Wall Street Journal corpus, in a way designed to record recurrent information over a wide range of size and terms of the description.

Extracting Partial Structures The database of partial structures is built up from the words in the corpus, by successively adding larger structures, after augmenting the corpus with the analysis provided by an unsupervised parser. The larger structures found in this way are then entered into the permanent database of structures only if a relation recurs with a frequency above a given threshold. When a structure does not meet the frequency threshold, it is generalized until it does.

The descriptive relationships admitted include:

- basic lexical features
  - spelling
  - part-of-speech
  - lemma
  - major category (maximal projection)
- dependency relations - depends on
- adjacency relations - precedes

Consider an example from the following sentence from the a training corpus of 20 million words of the Wall Street Journal.

(1) *Reserve board rules have put banks between a rock and a hard place*

The first order description of a word consists of its basic lexical features, i.e. the word spelling, its part of speech, its lemma, and its major category. Looking at the word *banks*, we have as description

TERMINAL
banks,NN,bank/N,NP

At the first level we add adjacency and dependency information, specifically

ADDED STRUCTURE
(precedes (put,VB,put/V,VG) (banks,NN,bank/N,NG))
(precedes (banks,NN,bank/N,NG) (between,IN,between/I,PG))
(depends (put,VB,put/V,VG) (banks,NN,bank/N,NG))

Assuming that we require at least two instances for a partial description to be entered into the database, none of these three descriptions qualify for the database. Therefore we must abstract away, using an arbitrarily defined abstraction path. First we abstract from the spelling to the lemma. This move admits two relations (since they are now frequent enough)

PRUNED STRUCTURES
(precedes (put,VB,put/V,VG) (,NN,bank/N,NG))
(depends (put,VB,put/V,VG) (,NN,bank/N,NG))

The third relation is still too infrequent, so we further generalize to

(precedes (,NN,,NG) (between,IN,between/I,PG))

a relation which is amply represented (3802 occurrences).

The process is iterated, using the current abstracted description of each word, adding a level of description, then generalizing when below the frequency threshold. Since each level in elaborating the description adds information to each word, it can only reduce the counts, but never increase them. This process finds a number of recurrent partial structures, including *between a rock and a hard place* (3 occurrences in 20 million words), and $[_{VP}put[_{NP}distance][_{PP}between]]$ (4 occurrences).

General Caveats There is of course considerable noise introduced by the errors in analysis that the parser makes.

There are several arbitrary decisions made in collecting the database. The level of the threshold is arbitrarily set at 3 for all structures. The sequence of generalization is arbitrarily determined before the training. And the predicates in the description are arbitrarily selected. We would like to have better motivation for all these decisions.

It should be emphasized that while the set of descriptive terms used in the collection of the partial structure database allows a more flexible description of the corpus than simple ngrams, CFG's or some dependency descriptions, it nevertheless is also restrictive. There are many predictive relationships that can not be described. For example, parallelism, reference, topic-based or speaker-based variation, and so on.

Motivation The underlying reason for developing a database of partial trees is not primarily for the language modeling task of predicting the next word. Rather the partial-tree database is motivated by the intuition that partial trees are are the locus of other sorts of linguistic information, for example, semantic or usage information. Our use of language seems to involve the composition of variably sized partially described units expressed in terms of a variety of predicates (only some of which are included in our database). Which units are selected in using language depends on a variety of factors, including meaning, subject matter, speaking situation, style, interlocutor and so on. Of course, demonstrating that this intuition is valid remains for future work.

The set of partial trees can be used directly in an analogical parser, as described in Hindle 1992. In the parser, we are not concerned with estimating probabilities, but rather with finding the structure which best matches the current parser state, where a match is better the more specific its description is.

3. ENHANCING A TRIGRAM MODEL

The partial structure database provides more information than an ngram description, and thus can be used to enhance an ngram model. To explore how to use the best available information in a language model, we turn to a trigram model of Wall Street Journal text. The problem is put into relief by focusing on those cases where the trigram model fails, that is, where the observed trigram condition (w_{-2}, w_{-1}) does not occur in the training corpus.

In the current test, we randomly assigned each sentence from a 2 million word sample of WSJ text to either the test or training set. This unrealistically minimizes the rate of unseen conditions, since typically the training and test are selected from disjoint documents (see Church and Gale 1991). On the other hand, since the training is only a million words, the trigrams are undertrained. In general, the rate of unseen conditions will vary with the domain to be modeled and the size of training corpus, but it will not (in realistic languages) be eliminated. In this test, 26% (258665/997811) of the bigrams did not appear in the test, and thus it is necessary to backoff from the trigram model.

We will assume that a trigram model is sufficiently effective at prediction in those cases where the conditioning bigram has been observed in training, and will focus on the problem of what to do when the conditioning bigram has not appeared in the training. In a standard backoff model, we would look to estimate $Pr(w_0|w_{-1})$. Here we want to consider a second predictor derived from our database of partial structures. The particular predictor we use is the lemma of the word that w_{-1} depends on, which we will call $G(w_{-1})$. In the example discussed above, the first (standard) predictor for the word *between* is the preceding word *banks* and the second predictor for the word *between* is $G(banks)$, which in this case is put/V.

We want to choose among two predictors, w_{-1} and $G(w_{-1})$. In general, if we have two conditions, C_a and C_b and we want to find the probability of the next word given these conditions. Intuitively, we would like to choose the predictor C_i for which the predicted distribution of w differs most from the unigram distribution. Various measures are possible; here we con-

model	logprob
unigram	9.55
backoff w_{-1}	8.06
backoff $G(w_{-1})$	8.20
backoff w_{-1} then $G(w_{-1})$	7.97
backoff (MAX IS of w_{-1} and $G(w_{-1})$)	7.99

Table 4: Backoff for unknown trigrams in WSJ text.

sider one, which Resnik (1993) calls *selectional preference*, namely the relative entropy between the posterior distribution $Pr(w|C)$ and the prior distribution $Pr(w)$. We'll label this measure *IS*, where

$$IS(w;C) = \sum_{w} Pr(w|C) log \frac{Pr(w|C)}{Pr(w)}$$

In the course of processing sentence (1), we need an estimate of $Pr(between|put\ banks)$. Our training corpus does not include the collocation *put banks*, so no help is available from trigrams, therefore we backoff to a bigram model, choosing the bigram predictor with maximum IS. The maximum IS is for *put/V* ($G(w_{-1})$) rather than for w_{-1} (*banks*) itself, so $G(w_{-1})$ is used as predictor, giving a logprob estimate of -10.2 rather than -13.1.

The choice of $G(w_{-1})$ as predictor here seems to make sense, since we are willing to believe that there is a complementation relation between *put/V* and its second complement *between*. Of course, the choice is not always so intuitively appealing. When we go on to predict the next word, we need an estimate of $Pr(a|banks\ between)$. Again, our training corpus does not include the collocation *banks between*, so no help is available from trigrams. In this case, the maximum IS is for *banks* rather than *between*, so we use *banks* to predict *a* rather than *between*, giving a logprob estimate of -5.6 rather than -7.10.

Overall, however, the two predictors can be combined to improve the language model, by always choosing the predictor with higher IS score.

As shown in Table 4, this slightly improves the logprob for our test set over either predictor independently. However, Table 4 also shows that a simple strategy of chosing the raw bigram first and the $G(w_{-1})$ bigram when there is no information available is slightly better. In a more general situation, where we have a set of different descriptions of the same condition, the IS score provides a way to choose the best predictor.

4. CONCLUSION

Recurrent structures in text vary widely both in size and in the terms in which they are described. Existing grammars are too restrictive both in the size of structure they admit and in their terms of description to adequately capture the variation in text. A method has been described for collecting a database of partial structures from text. Methods of fully exploiting the database for language modeling are currently being explored.

5. REFERENCES

1. Church, Kenneth W., William A. Gale, Patrick Hanks, and Donald Hindle. 1991. "Using statistics in lexical analysis." in Uri Zernik (ed.) *Lexical acquisition: using on-line resources to build a lexicon*, Lawrence Erlbaum, 115-164.
2. Church, Kenneth W. and William A. Gale. 1991. "A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams," *Computer Speech and Language*, 5, 19-54.
3. Hindle, Donald. 1992. "An analogical parser for restricted domains," In *Proceedings of the Fifth DARPA Workshop on Speech & Natural Language*, -.
4. Hindle, Donald. 1981. "A probabilistic grammar of noun phrases in spoken and written English," In David Sankoff and Henrietta Cedergren (eds.) *Variation Omnibus*, Linguistic Research, Inc. Edmonton, Alberta.
5. Melchuk, Igor A. 1988. *Dependency Syntax: Theory and Practice*, State University of New York Press, Albany.
6. Resnik, Philip. 1993. "Semantic Classes and Syntactic Ambiguity," This volume.
7. Schabes, Yves. 1988. "Parsing strategies with 'lexicalized' grammars: application to tree adjoining grammars", in Proceedings fo the 12th International Conference on Computational Linguistics, COLING88, Budapest, Hungary.