# Valence Induction with a Head-Lexicalized PCFG

**Glenn Carroll and Mats Rooth**
IMS, Universität Stuttgart
{glenn,mats}@ims.uni-stuttgart.de

## Introduction

Either directly or indirectly, the lexicon for a natural language specifies *complementation frames* or *valences* for open-class words such as verbs and nouns. Constructing a lexicon of complementation frames for large vocabularies constitutes a challenge of scale, with the further complication that frame usage, like vocabulary, varies with genre and undergoes ongoing innovation in a living language. This paper addresses this problem by means of a learning technique based on probabilistic lexicalized context free grammars and the expectation-maximization (EM) algorithm. Given a hand-written grammar and a text corpus, frequencies of a head word accompanied by a frame are estimated using the inside-outside algorithm, and such frequencies are used to compute probability parameters characterizing subcategorization. The procedure can be iterated for improved models. We show that the scheme is practical for large vocabularies and accurate enough to capture differences in usage, such as those characteristic of different domains.

## A grammar and formalism

The core of the grammar is an $\bar{X}$ grammar (Jackendoff [1977]) of phrases including noun phrases, prepositional phrases, and verbal clusters. A representative verbal structure is given on the left in Figure 1. The symbol VFC is read "finite verb chunk"; similarly we work with noun chunks (NC), prepositional chunks (PC), and so forth. Our use of the chunk concept follows Abney [1991], Abney [1995]. Categories are interpretable in terms of a feature decomposition, but are treated as atomic in the formalism. We depart from a standard context-free formalism in that heads are marked on the right hand sides of rules, using a prime (').

The grammar includes complementation rules for verbs, nouns, and adjectives. Complements are attached at a level above the chunk, which we call the phrasal level. For instance, the category VFP is expanded as a finite verb chunk VFC and a sequence of complements. This is illustrated on the right in Figure 1, where the VFC headed by *decided* takes a VTOP complement, the VTOC headed by *emphasize* takes an NP complement, and so forth.

Finally, the least standard part of the grammar is a large set of *state* or *n-gram* rules which form a parse without constructing a standard clause-level analysis. Instead, phrasal categories are strung together with context-free rules modelling a finite state machine, where the states are categories consisting of an ordered pair of phrasal categories. This results in right-branching structures, as illustrated Figure 2. Note that the entire tree on the right in Figure 1 could be substituted for the finite verb phrase VFP in the tree on the left in Figure 2. The state rules allow almost all the sentences (about 97%) in the corpus to be parsed, at the price of not assigning linguistically realistic higher-level structure.

We now define headed context-free grammars in the sense employed here.

**Definition.** A headed context free grammar is a tuple $\langle N, T, W, \mathcal{L}, \mathcal{R}, s\rangle$, where: (i) $N$ and $T$ are disjoint sets, interpreted as the non-terminal and terminal categories respectively. (ii) $W$ is a set, interpreted as the set of words. (iii) $\mathcal{L}$ is a relation between $W$ and $T$, indicating the possible terminal categories (parts of speech) for a given word. (iv) The set of headed productions $\mathcal{R}$ is a finite subset of $N \times N^* \times (N \cup T) \times N^*$, such that each non-terminal occurs as the left hand side of some rule and each terminal occurs on the right hand side of some rule. (v) $s \in N$, with the interpretation of a start symbol.

We typically use $\bar{n}$ as a variable for mother categories, $n$ for head daughter categories, and $\alpha$ and $\beta$ for the category sequences flanking the head on the right hand side, so that $\langle\bar{n}, \alpha, n, \beta\rangle$ represents a rule. $x$ is used as a variable for non-head categories.

A category $\bar{n}$ in $N$ is a *projection* of a category $n$ in $N \cup T$ if there is some rule of the form $\langle\bar{n}, \alpha, n, \beta\rangle$. The set of *lexicalized nonterminals* $\mathcal{N} \subseteq W \times N$ is the composition of $\mathcal{L}$ with the transitive closure of the

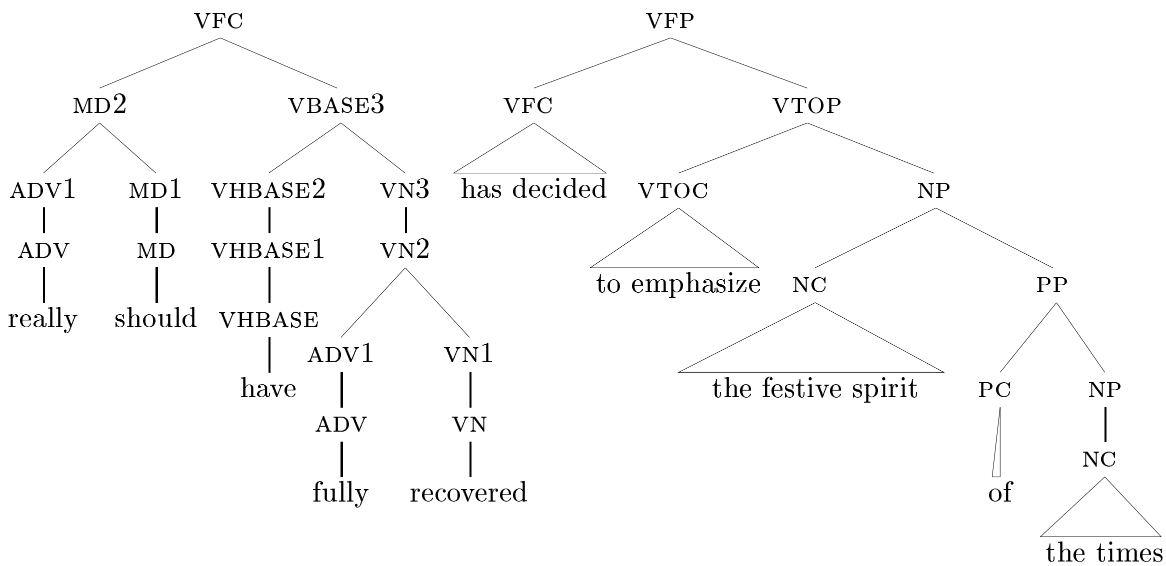



Figure 1: Illustrations of a finite verb chunk and complementation.

projection relation. We have $\langle w, n\rangle \epsilon \mathcal{N}$ if the word $w$ can be the lexical head of the nonterminal category $n$ (in a complete or incomplete tree).

## Lexicalization and the probability model

This section defines a parameterized family of probability distributions over the trees licensed by a head-lexicalixed CFG. The main ideas on the parameterization of a lexicalized context free grammar which are employed here derive from Charniak [1995]; see also the remarks on lexicalization in Charniak [1993, section 8.4].

The head marking on rules is used to project lexical items up a chain of categories. In the transitive verb phrase on the right in Figure 2, *question* is projected to the NP level, and *asked* is projected to the VFP level. In this tree, the non-terminal nodes are lexicalized non-terminals, while the terminal nodes are members of $\mathcal{L}$. The point of projecting head words is to make information which probabilistically conditions rules and lexical choices available at the relevant level. At the top level in this example, the head *asked* is used to condition the choice of the phrase structure rule VFP $\rightarrow$ VFC$'$ NP as well as the choice of *question*, the head of the object.

We now define events which characterize choices of rules and of lexical heads.

**Definition.** Given a grammar $G = \langle N, T, W, \mathcal{L}, \mathcal{R}, s\rangle$ with lexicalized non-terminals $\mathcal{N}$, the set of rule events $ER(G)$ is the set of tuples $\langle w, \bar{n}, \alpha, n, \beta\rangle$ such that $\langle w, \bar{n}\rangle$ is an element of $\mathcal{N}$ and $\langle \bar{n}, \alpha, n, \beta\rangle$ is an element of $\mathcal{R}$. The set of lexical choice events $EL(G)$ is the set of tuples $\langle w, \bar{n}, x, v\rangle$ such that (i) $\langle w, \bar{n}\rangle$ and $\langle v, x\rangle$ are elements of $\mathcal{N}$;[1] (ii) in some rule of the form $\langle \bar{n}, \alpha, n, \beta\rangle$, $x$ is an element of one or both of the category sequences $\alpha$ and $\beta$; and

By virtue of the length of tuples, $ER(G)$ and $EL(G)$ are disjoint, and the union $E(G)$ can be formed without confusing lexical with rule events.

A head-lexicalized PCFG is represented as a function mapping events to real numbers.

**Definition.** Let $G$ be a headed context free grammar. A head-lexicalized probabilistic context free grammar with signature $G$ is a function $p$ with domain $E(G)$ and range $[0, 1]$ satisfying the conditions: (i) Fixing any lexicalized non-terminal $\langle \bar{w}, \bar{n}\rangle$, $\sum_{\alpha,n,\beta} p_{\bar{w},\bar{n},\alpha,n,\beta} = 1$; (ii) Fixing any lexicalized non-terminal $\langle \bar{w}, \bar{n}\rangle$ and possible non-head daughter $x$, $\sum_{x,w} p_{\bar{w},\bar{n},x,w} = 1$. Here the value of the function $p$ on a rule event is written as $p_{\bar{w},\bar{n},\alpha,n,\beta}$, and on a lexical event as $p_{\bar{w},\bar{n},x,w}$.

To assign probability weights to trees, we use a tree-licensing and labelling interpretation of the grammar; a node in a tree analysis is labeled with event corresponding to the rule used to expand the node, and the list of lexical events for the non-head daughters of the node. Where $\tau$ is a labeled tree li-

[1] In the events, conditioning factors are ordered in the way they are dropped off in the smoothing procedure described below. In a lexical event $\langle w, \bar{n}, x, v\rangle$, the choice of the word $v$ is conditioned on the parent lexical head $w$, the parent category $\bar{n}$, and the child category $x$. In the first smoothing distribution, the first conditioning factor, i.e. the parent head $w$, is dropped.

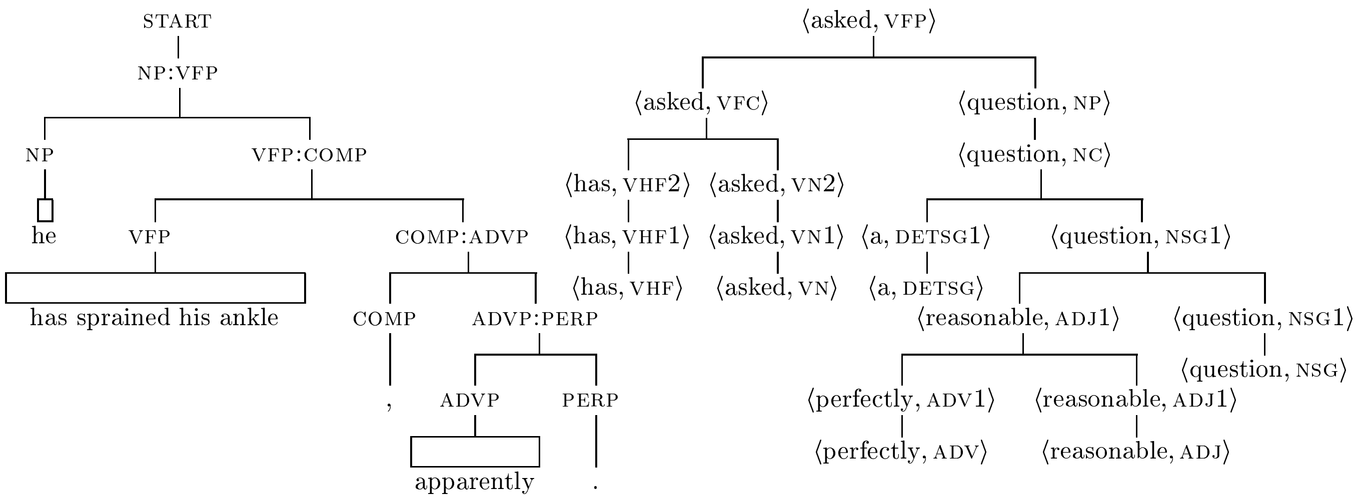



Figure 2: Left: finite-state structure; Right: Lexicalization.

censed by G, we define $e(\tau) : E(G) \to \mathbb{N}$ to be a function counting occurrences of events as labels in $\tau$. Algebraically, we think of $e(\tau)$ as a monomial in the variables $E(G)$; the exponent of a given variable (or event) $z$ is the number of occurrences of $z$ in $\tau$. We denote the evaluation of a polynomial or monomial $\phi$ in the variables $E(G)$ by subscripting: $\phi_p$ is the value of $\phi$ at the vector of reals $p$. Relative to a parameter setting $p$, $[e(\tau)]_p$ is interpreted as the probabilistic weight of the labeled tree $\tau$.[2]

These notions are exemplified in Figure 3, which is a phrase structure tree for the N1 (read: N-bar) *big big problem* in a grammar where N1 is the sentence category. Each non-terminal is labeled with a phrase structure rule, and with lexical choice events for non-head daughters. In this case, the only non-head daughters are the two A1's headed with head *big*. ⟨problem,N1,A1,big⟩ is a lexical choice event where *big* is selected as the head of an A1 with parent category N1, and parent head *problem*. An event monomial corresponding to the event tree is obtained as the symbolic product of the events labeling the tree.

## Parameter Estimation

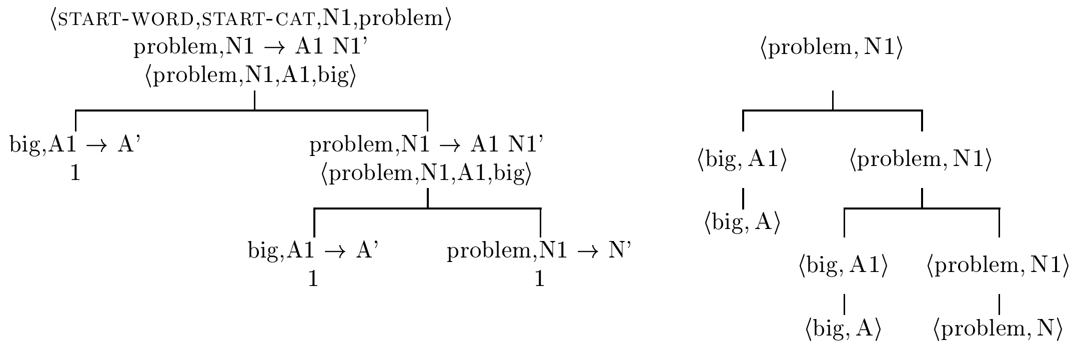
Given a grammar $G$, the inductive problem is to estimate a head-lexicalized PCFG with signature $G$. We work with the standard method for estimating PCFGs, based on the Expectation-Maximization framework (Baum & Sell [1968]; Dempster, Laird & Rubin [1977]).

Above, we defined the event polynomial $e(\tau)$ for an event tree $\tau$ licensed by $G$. The event polynomial for a sentence $\sigma$ is the sum of the event polynomials for the event trees with yield $\sigma$. Where corpus $\mathcal{C}$ is a sequences of sentences, the corpus event polynomial $e(\mathcal{C})$ is the (polynomial) product of the event polynomials for the sentences in $\mathcal{C}$. In these terms, maximum likelihood estimation selects a parameter setting $p$ such that the value $[e(\mathcal{C})]_p$ of the corpus polynomial is maximized; this corresponds to selecting a parameter setting which maximizes the probability of the corpus.

The E step of the EM algorithm computes an expected event count function which can be defined in terms of the corpus polynomial. In the estimation of PCFGs using the inside-outside algorithm, event counts are computed iteratively, sentence by sentence. The computation uses a packed parse forest, a compact and-or graph representing a set of trees and the sentence event polynomial, and which allows efficient computation of expected event counts. Somewhat more formally, we use the Inside-outside algorithm (Baker [1979]). to compute $E_p(z|\sigma) : E(G) \to \mathbb{R}$ where $z$ ranges over events in the join rule and lexical event space $E(G)$, defined earlier. $c(\sigma, p)(z)$ has the probabilistic interpretation of the expected number of occurrences of the event $z$ in the set of trees with yield $\sigma$.

Given a parameter setting $p$, event counts are computed and summed over the sentences in the corpus. In the algorithm of Baum and Sell, new parameter values would be defined as relative frequencies of event counts, i.e. maximum-likelihood estimation based on hidden data in the EM framework. We

[2] As with ordinary PCFGs, depending on the parameters, the construction may or may not define a probability measure on the set of finite trees licensed by $G$. For the general case, infinite trees can be included in the sample space. This requires an extension in the definition of the measure but does not affect the probabilities of finite trees.




$$(\text{problem,N1} \rightarrow \text{A1 N1'})^2 \langle \textsc{start-word},\textsc{start-cat},\text{N1,problem}\rangle^1 (\text{big,A1} \rightarrow \text{A'})^2 \langle \text{problem,N1,A1,big}\rangle^2 (\text{problem,N1} \rightarrow \text{N'})^1$$

Figure 3: On the left, an event tree. On the right, the corresponding lexicalized tree. On the bottom, the event monomial obtained as a symbolic product of the event labels. The lexical choice event involving START-CAT chooses the head of the sentence, in this case *problem*.

use instead a modified M step involving a smoothing scheme in order to deal with the size of the parameter space and the resulting problems that (i) counts are zero for the majority of events, and (ii) the parameter space is too large to be represented directly in computer memory. Lexicalized rules are smoothed against non-lexicalized rules in a standard back-off scheme (Katz [1980]). The smoothed probability is defined as a weighted sum of the maximum-likelihood estimates for the lexicalized and unlexicalized rule probabilities. The smoothing weight is allowed to vary through five discrete values as a function of the frequency of the word-category pair. The parameters give greater weight to the lexicalized distribution when enough data is present to justify it. The smoothing parameters are set using the EM algorithm on reserved data.

For the lexical choice distributions, an absolute discounting scheme from Ney, Essen & Kneser [1994] is used, which is similar to Good-Turing, but somewhat simpler to work with.

## The experiment

We estimated a head-lexicalized PCFG from parts of the British National Corpus (BNC Consortium [1995]), using the grammar described in the first section and the estimation method of the previous section. A bootstrapping method was used, in which first a non-lexicalized probabilistic model was used to collect lexicalized event counts. On the next iteration, counts were estimated based on a lexicalized weighting of parses, as described in the previous section.

Analyses were restricted to those consistent with the part of speech tags specified in the BNC, which are produced with a tagger. In each lexicalized iteration, event counts were collected over a contiguous five million word segment of the corpus. Parameters were re-computed in the way described above, and the procedure was iterated on the next contiguous five-million word segment. Results from all iterations were pooled to form a single model estimated from 50M words. Table 1 illustrates lexical distributions in this model.

This training scheme allows the frame distributions for high-frequency words a chance to converge on their true distributions, whereas a single 50M word iteration would not. The strategy derives from a variant generalized EM algorithm presented in Neal & Hinton [1998]. In a nutshell, re-estimating the parameters during the course of a single training iteration will still lead to convergence on a maximum-likelihood estimate, provided certain conditions are met. Foremost among these is the requirement that no parameter setting can be prematurely set to zero; this is met by our smoothing strategy. This is not to say that precisely the same strategy, pursued across multiple iterations, would produce a maximum-likelihood estimate; it would not. However, "classical" EM, requiring repeated iteration over the entire training set, is both relatively inefficient and infeasible given our present computational resources.

### Dictionary Evaluation

The comparison to frames specified in a dictionary we use was introduced by Brent [1993] and subsequently used by Manning [1993], Ersan & Charniak [1995] and Briscoe & Carroll [1996]. The measure uses *precision* and *recall* to compare the set of induced frames to those in the standard. Precision is the percentage of frames that the system proposes that are correct (i.e. in the standard). Recall is the percentage of frames in the standard that the system

| $p_{\mathrm{NP},satisfactory,\mathrm{ADJP},w}$ | | $p_{\mathrm{VFP},address,\mathrm{NP},w}$ | |
|---|---|---|---|
| adverb | prob | noun | prob |
| entirely | 0.17 | question | 0.086 |
| highly | 0.11 | issue | 0.086 |
| most | 0.09 | themselves | 0.059 |
| very | 0.075 | issues | 0.031 |
| quite | 0.055 | structure | 0.031 |
| wholly | 0.032 | argument | 0.014 |
| uncommonly | 0.0037 | questions | 0.0043 |
| especially | 0.0037 | electorate | 0.0043 |
| ... | | ... | |

Table 1: On the left: the eight largest parameters in the lexical choice distribution describing modifying adjectives selected by *satisfactory*. On the right: parallel information for the distribution describing heads of objects of the verb *address*.

proposes. If the results are broken down into true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), precision is defined as $TP/(TP+FP)$ and recall is $TP/(TP+FN)$. To produce measurements from our system, we must first reduce our distributions to set membership. Brent proposed a stochastic filter for this reduction, consisting of a set of per-frame probability cutoffs, which are applied independently of the lexical head. Although though the independence assumption is certainly dubious, we have adopted this method, without change, except for the introduction of a heuristic for finding the frame cutoffs.

The key property of cutoffs is that they control the tradeoff of precision versus recall. Raising the cutoff will generally produce a higher precision, but lower recall, and contrariwise. As we are neutral about this tradeoff, we set the cutoffs at the crossover point, where the difference in precision and recall changes sign. This is not entirely deterministic, as the measures may cross more than once; in that case, we optimize for the best precision.

For our dictionary, we used *The Oxford Advanced Learner's Dictionary* (Hornby [1985]), also used by Ersan/Charniak and Manning. We reduced our frame set and the dictionary's to a common set, mapping some frames and eliminating others. For evaluation, we selected 200 verbs at random from among those that occurred more than 500 times in the training data; half were used to set the optimal cutoff parameters, and precision and recall were measured with the remainder.

Table  shows results broken down by frame. The largest source of error is the intransitive frame. It is not hard to understand why: our robust parsing architecture resolves unparsable constructs as intransitives. In addition to sentences where verbs are not

| | cutoff | TP | FP | FN | prec | recl |
|---|---|---|---|---|---|---|
| Intrans | 0.15 | 20 | 24 | 12 | 0.6471 | 0.7857 |
| NP | 0.021 | 3 | 5 | 1 | 0.9479 | 0.9891 |
| ADJ | 0.079 | 92 | 0 | 6 | 1 | 0.25 |
| PP | 0.045 | 27 | 15 | 6 | 0.7761 | 0.8966 |
| PART | 0.027 | 60 | 5 | 14 | 0.8077 | 0.6 |
| VTOP | 0.079 | 83 | 1 | 7 | 0.9 | 0.5625 |
| NP PP | 0.040 | 26 | 11 | 10 | 0.8281 | 0.8413 |
| NP PART | 0.0099 | 68 | 6 | 12 | 0.7 | 0.5385 |
| NP NP | 0.036 | 81 | 6 | 8 | 0.4545 | 0.3846 |
| NP VTOP | 0.018 | 84 | 1 | 6 | 0.9 | 0.6 |
| VING | 0.019 | 86 | 3 | 6 | 0.625 | 0.4545 |
| NP VING | 0.017 | 93 | 3 | 2 | 0.4 | 0.5 |
| NP VINF | 0.019 | 99 | 1 | 0 | 0 | – |
| NP ADJ | 0.016 | 85 | 1 | 12 | 0.6667 | 0.1429 |
| PP VTOP | 0.014 | 97 | 1 | 1 | 0.5 | 0.5 |
| | | 310 | 83 | 103 | 0.7888 | 0.7506 |

Table 2: Precision/recall broken down by frame.

linked up with their complements because of interjections, complex conjunctions or ellipses, this includes frames such as SBAR and WH-complements which are not included in the chunk/phrase grammar. While it would be possible in principle to extract these from the present word collocation statistics, we plan instead to pursue a solution involving extensions in the grammar.

A second major source of error is prepositional phrases. The complementation model embodied in the PCFG does not distinguish complements from adjuncts, and therefore adjunct prepositional phrases are a source of false positives. Thus the NP PP frame is scored as a false positive for the verb *meet*, because the OALD does not list the frame, although the combination appears often in the corpus data. While such frames lead to a loss of precision in the dictionary evaluation, we do not necessarily consider them a flaw in the information learned by the system, since the argument/adjunct distinction is often tenuous, and adjuncts are in many cases lexically conditioned.

Lastly, there are many false negatives for the particle frame and noun plus particle. This is mainly due to disagreements between BNC particle tagging and particle markup in the OALD.

Despite these difficulties, the summary shown in table  shows results that are on the whole favorable. In comparison with other work with a comparable number of frames (Manning, Ersan/Charniak), the system is well ahead on recall and well behind on precision. If one takes the sum of precision and recall to be the final performance indicator, than we are slightly ahead: 1.54 vs. 1.44 for Ersan and 1.33

| | precision% | recall % | no. of frames |
|---|---|---|---|
| lex PCFG | 79 | 75 | 15 |
| Briscoe | 66 | 36 | 159 |
| Charniak | 92 | 52 | 16 |
| Manning | 90 | 43 | 19* |

Table 3: Type precision/recall comparison. Some of Manning's frames are parameterized for a preposition.

for Manning. Briscoe and Carroll's work, with ten times as many target frames, is so different that the numbers may be regarded as incomparable.

Obviously, precision and recall measured against a standard relies on the completeness and accuracy of that standard. In checking false positives, Ersan and Charniak found that the OALD was incomplete enough to have a serious impact on precision. Symmetrically, false negatives conflate deficiencies in the corpus with poor learning efficiency. It is impossible to say based on table which of the systems is more efficient at learning. While our system shows the best recall, this could be attributed to our having the best training data. Charniak used 40M words of training data, comparable to our 50M, but his data was homogeneous, all taken from the Wall Street Journal. As we will show below, frame usage varies across genres, so the BNC, which includes texts from a wide variety of sources, shows more varied frame usage than the WSJ, and thus provides better data for frame acquisition.

## Cross entropy evaluation

The information-theoretic notion of *cross entropy* provides a detailed measure of the similarity of the acquired probabilistic lexicon to the distribution of frames actually exhibited in the corpus (which we call the empirical distribution). The cross entropy of the estimated distribution $q$ with the empirical distribution $p$ obeys the identity

$$CE(p, q) = H(p) + D(p\|q)$$

where $H$ is the usual entropy function and $D$ is the relative entropy, or Kullback-Leibler distance. The entropy of a distribution over frames can be conceptualized as the average number of bits required to designate a frame in an ideal code based on the given distribution. In this context, entropy measures the complexity of the observed frame distribution. The relative entropy is the penalty paid in bits when the frame is chosen according to the empirical distribution $p$, but the code is derived from the model's estimated distribution, $q$. Relative entropy is always non-negative, and reaches zero only when the two

| obs freq | | | est freq | |
|---|---|---|---|---|
| imag | natsci | frame | imag | natsci |
| 51 | 39 | NP VTOP | 40.4 | 34.2 |
| 21 | 43 | NP | 20.7 | 33.1 |
| 13 | 6 | NP NP | 8.8 | 3.9 |
| 6 | 1 | NP PP | 3.2 | 4.7 |
| 5 | 1 | NP PART | 1.7 | 1.0 |
| 2 | 11 | PP | 1.8 | 10.2 |
| 1 | 0 | SBAR | 0 | 0 |
| 1 | 0 | Intrans | 9.3 | 7.6 |
| 2.130 | 1.913 | entropy | 2.476 | 2.423 |

Table 4: True and estimated frame frequencies for *allow*.

distributions are identical. Our goal, then, is to minimize the relative entropy. For more in-depth discussion of entropy measures, see Cover & Thomas [1991], or any introductory information theory text.

For relative entropy to be finite, the estimated distribution $q$ must be non-zero whenever $p$ is. However, some observed frames are not present in the grammar, for one of two reasons. Some well-known frames such as SBAR require high-level constructs not available in the chunk/phrase grammar and unusual/unorthodox frames turn up in the data, e.g. PART PP PP. Since the model lacks these frames, smoothing against the unlexicalized rules is insufficient. Instead, for all the estimated distributions, we smooth against a Poisson distribution over categories, which assigns non-zero probability to all frames, observed or not. This allows us to spell out the unknown frame using a known finite alphabet, the grammar categories, while retaining a reasonable average length over frames.

For our entropy measurements, we selected three verbs, *allow*, *reach*, and *suffer* and extracted about 200 occurrences of each from portions of the BNC not used for training. Half of each sample was drawn from "imaginative" text and the other half from the natural or applied sciences, as indicated by BNC text mark-up. The true frame for each verb occurrence was marked by a human judge[3]. The empirical distribution was taken as the maximum-likelihood estimate from these frequencies. Tables 4 and 5 indicate the observed frequencies and the entropy of the resulting distributions.

Alongside the observed frequencies, we indicate a set of estimated frequencies. These were generated by taking the 50M word model described above, parsing the test sentences, and extracting the estimated frequencies. The sum of estimated frequencies is gen-

[3] For this judgment, the frame set was unrestricted, i.e. included frames not in the grammar.

| obs freq | | | est freq | |
|---|---|---|---|---|
| imag | natsci | frame | imag | natsci |
| 63 | 88 | NP | 50.1 | 74.5 |
| 13 | 15 | NP PP | 5.9 | 10.9 |
| 9 | 1 | PART | 5.9 | 0.8 |
| 6 | 0 | PART PP | 2.7 | 0 |
| 5 | 3 | PP | 6.7 | 3.4 |
| 4 | 1 | Intrans | 15.2 | 6.8 |
| 2 | 0 | PART NP | 0.5 | 0 |
| 1 | 0 | NP PART | 0 | 0.1 |
| 2.0 | 0.979 | entropy | 2.101 | 1.473 |

| obs freq | | | est freq | |
|---|---|---|---|---|
| imag | natsci | frame | imag | natsci |
| 41 | 6 | Intrans | 34.9 | 13.4 |
| 31 | 54 | PP | 27.4 | 50.5 |
| 21 | 36 | NP | 18.9 | 23.0 |
| 4 | 1 | NP VTOP | 2.1 | 0.7 |
| 3 | 4 | NP PP | 0.9 | 5.2 |
| 1.936 | 1.580 | entropy | 1.936 | 1.907 |

Table 5: True and estimated frame frequencies for *reach* (top) and *suffer* (bottom).

| | | | $D(p\|q)$ for various $q$ | | |
|---|---|---|---|---|---|
| *head, genre* | | $H(p)$ | other genre | 50M lex | 50M unlex |
| allow | imag | 2.06 | 0.50 | 0.40 | 3.13 |
| | natsci | 1.78 | 0.49 | 0.42 | 2.27 |
| reach | imag | 1.99 | 0.91 | 0.35 | 1.07 |
| | natsci | 0.90 | 0.37 | 0.37 | 1.36 |
| suffer | imag | 1.86 | 0.87 | 0.24 | 0.70 |
| | natsci | 1.51 | 0.59 | 0.37 | 1.19 |
| | mean | 1.68 | 0.62 | 0.36 | 1.62 |

Table 6: Frame relative entropy for three verbs in two genres. The first column names the lexical head and genre, and the second the entropy ($H$) of the empirical distribution over frames, $p$. By empirical distribution we mean the relative frequencies from examples scored by a human judge. Columns three through five give the relative entropy $D(p\|q)$ for various related distributions. In column three, $q$ is the empirical frame distribution for the same head, but with the complementary genre. In column four $q$ is the (genre-independent) distribution derived from the 50M word lexicalized model. Column five uses the unlexicalized frame distribution derived from the 50M model, i.e. a distribution insensitive to the head verb. Lower relative entropy is better.

erally less than the observed frequencies due to tagging errors, parse failures, and frequency assigned to frames not shown in the tables. However, an eyeball inspection of the tables shows that the parser does a good job of reproducing the target distribution.

One striking feature in the tables is the variation across genre. In particular, *suffer* used in the imaginative genre shows a very different distribution than *suffer* in the natural sciences. A chi-squared test applied to each pair indicates that the samples come from distinct distributions (confidence $> 95\%$).

The column labeled "50M lex" in Table 6 provides a quantitative measure of the agreement between the 50M word combined model and the empirical distributions for the three verbs in two genres in the form of relative entropy. The first column repeats the entropy of the data distributions. For purposes of comparison, the second column indicates the relative entropy of one data distribution with the other data distribution filling the role of the estimated distribution (i.e. $q$) in the discussion above. The relative entropy is lower when the estimated distribution is used for $q$ than when the data distribution for the other genre is used for $q$ in each case but one, where the figures are the same. This suggests the combined model contains fairly good overall distributions.

To numerically evaluate whether the system was able to learn the distribution exhibited in a given collection of sentences, we tuned the lexicon by parsing the test sentences for each genre separately with the 50M word model, extracting the frequencies, and estimating the distribution from these. The results are the column 4 labeled "50M lexicalized extraction" in 7. The following columns give the same figures for freqency extraction with other models. Extraction with the large lexicalized model gives the best results, and gives better relative entropy than the 50M lexicalilazed model itself (in column 2). Notice that only the distributions estimated with the two 50M models are better than the 50M lexicalized model, though the unlexicalized one is only marginally better. In this sense, only the 50M lexicalized parser proves to be a good enough parser for genre tuning. Notice that with this model, tuning in no case gives worse relative entropy, and in five out of six cases give an improvement.

Notice also that relative entropy for the distributions obtained by tuning with the 50M model are a good deal lower than the cross-genre figures from Table 6. This suggests that if we wanted to have a good probabilistic lexicon for, say, the imaginative genre, we would be better off using the automatic extraction procedure on data drawn from that genre than using a *perfect* parser (or a lexicographer) on data drawn from some other genre, such as the natural sciences. This provides a calibration of the accuracy of the lexicalized parser's estimates, and conversely demonstrates that words are not used in the same

| | $D(p \| q)$ | | | | |
|---|---|---|---|---|---|
| *head, genre* | 50M lex mod | 50M lex extr | 5M lex extr | 50M unl. extr | 5M unl. extr |
| allow imag | 0.40 | 0.32 | 1.32 | 0.47 | 1.32 |
| allow natsci | 0.42 | 0.28 | 0.28 | 0.52 | 0.86 |
| reach imag | 0.35 | 0.35 | 0.63 | 0.32 | 0.63 |
| reach natsci | 0.37 | 0.19 | 0.34 | 0.28 | 0.34 |
| suffer imag | 0.24 | 0.11 | 0.38 | 0.12 | 0.38 |
| suffer natsci | 0.37 | 0.20 | 0.88 | 0.34 | 0.88 |
| mean | 0.36 | 0.24 | 0.64 | 0.34 | 0.74 |

Table 7: Relative entropy of distributions estimated by parsing the test sentences with various models, and using the Inside-outside algorithm to produce estimated distributions $q$. The first column names empirical distributions $p$. The second column repeats relative entropy for the 50M lexicalized model from the previous table. The third gives relative entropy where $q$ is obtained by parsing and estimating frequencies in the test sentences with the 50M lexicalized model. The following columns give the corresponding figures for a $q$ obtained by following the same procedure with a 5M word lexicalized model, a 50M word unlexicalized model, and a 5M word unlexicalized model.

way in different genres.

## Optimal parses

Although identifying a unique parse does not play a role in our experiment, it is potentially useful for applications. A simple criterion is to pick a parse with maximal probability; this is identified in a parse forest by iterating from terminal nodes, multiplying child probabilities and the local node weight at *and*-nodes (chart edges), and choosing a child with maximal probability at *or*-nodes (chart constituents). Figures 1 and 4 give examples of maximal probability probability parses.

Other optimality criteria can be defined. The structure on noun chunks is often highly ambiguous, because of bracketing and part of speech ambiguities among modifiers. For many purposes, the internal structure of an noun chunk is irrelevant; one just wants to identify the chunk. From this point of view, a probability estimate which considers just one analysis might underestimate the probability of a noun chunk. In what we call a sum-max parse, probabilities are summed within chunks by the inside algorithm. Above the chunk level, a highest-probability tree is computed, as described above.

## Notes on the implementation and parsing times

Software is implemented in C++. The parser used for the bootstrap phase is a vanilla CFG chart parser, operating bottom-up with top-down predictive filtering. Chart entries are assigned probabilities using the unlexicalized PCFG, and the lexicalized frequencies are found by carrying out a modified inside-outside algorithm which simulates lexicalization of the chart.

In the iterative training phase, an unlexicalized context-free skeleton is found with the same parser. We transform this into its lexicalized form—categories become $\langle w, n \rangle$ pairs and rules acquire lexical heads—and carry out the standard inside-outside using the more elaborate head-lexicalized PCFG model. Average speed of the parser during iterative training, including parsing, probability calculation, and recording observations, is 10.4 words per second on a Sun SPARC-20. The memory requirements for a model generated from a 5M word segment are about 90Mbyte. The upshot of all this is that we can train about 1M words per day on one machine, and a single 5M word iteration requires one machine work week.

## Discussion

We believe the formalism and methodology described here have the following advantages:

- The grammar is under the control of the computational linguist and is of a familiar kind, making it possible to incorporate standard linguistic analyses, and making results interpretable in terms of linguistic theory. In contrast, approaches where context free rules are learned are likely to produce structures which are uninterpretable in terms of linguistic theory and practice.
- Because of the context free framework, efficient parsing algorthims (chart parsing) and probabilistic algorithms (the inside-outside algorithm) can be applied. With an efficient implementation, this makes it possible to construct representations of all the tree analyses for the sentences in corpora on the scale of ten to a hundred million words, and to map such a corpus to a probabilistic lexicon.
- With the robustness introduced by the state model, almost all sentences in the corpus can be parsed.
- The model assigns probabilities to sentences and trees, which is useful for applications independent of the lexicon-induction problem discussed here.
- The word-selection model, which threads a word bigram model through head relations in the syntactic tree, allows a large body of word-word collocations to be learned from the corpus, and put to use in weighting of competing analyses.

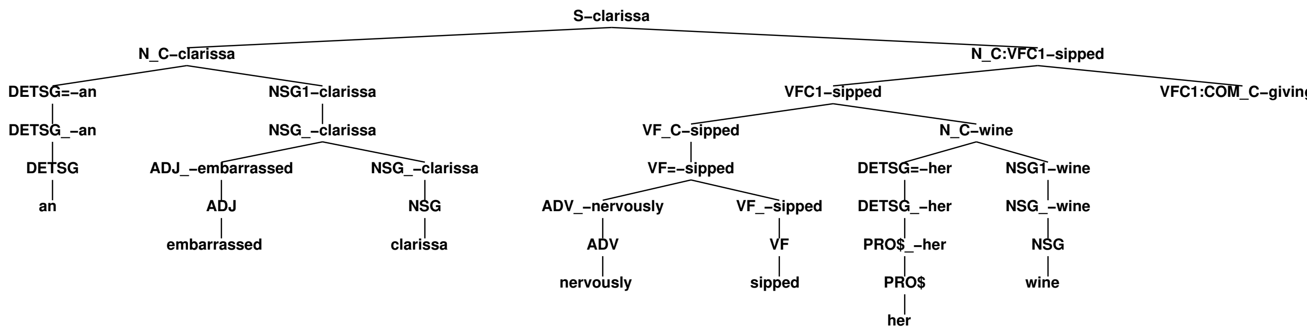



Figure 4: The first part of maximum probability parse.

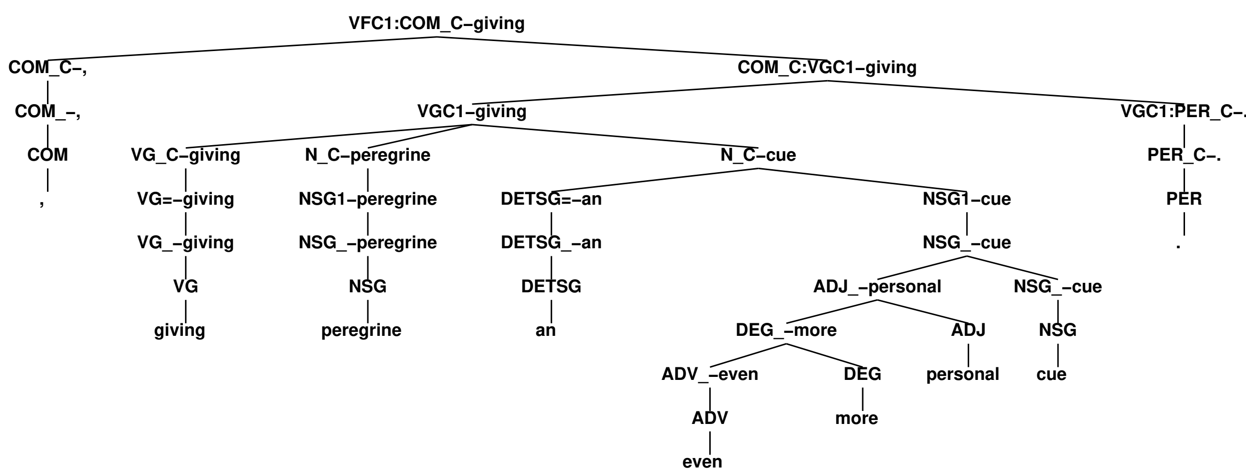



Figure 5: The second part.

- The valence information learned, rather than being simply a set of subcategorization frames, is a probability distribution which reflects the freqency of frames in a given training sample, and which can be plugged back into the parser and used to analyze further text.

Some of these benefits are purchased at the cost of a lack of sophistication in the grammar formalism, compared to constraint-based formalisms used in contemporary computational linguistics. This compromise is made in order to make large-scale experiments achievable; our interest is in conducting scientific experiments—observational and modeling experiments—with large bodies of language use. It is natural that this should require incorporating approximations in computational models. Notably, the compromises made in our approach are not so severe that the grammatical analyses identified and the probability parameters learned are out of touch with linguistic reality. This is in contrast to the situation with other approaches using similar mathematical methods, such as terminal-string n-gram modeling.

## Conclusion

We have presented a statistically-based method for valence induction, based on the idea of automatic tuning of the probability parameters of a grammar. On the standard precision/recall measures, our system performs better on precision, worse on recall, and on the whole somewhat better than other published systems. We have provided a more precise evaluation via entropy measures, showing that the model learns efficiently and builds accurate models of frame distributions. The cross-domain entropy of the data frame distributions provides numerical evidence that frame usage varies across domains, similar to word usage. This, in turn, suggests that automatic acquisition and stochastic tuning are a must for large-scale NLP applications and computational linguistic models.

## Bibliography


Abney, S. [1991], "Parsing by Chunks," in *Views on Phrase Structure*, D. Bouchard & K. Leffel, eds., Kluwer Academic Publishers.

——— [1995], "Chunks and dependencies: Bringing processing evidence to bear on syntax," in *Linguistics and Computation*, Jennifer S. Cole, Georgia M. Green & Jerry L. Morgan, eds., CSLI Publications.

BNC Consortium [1995], *The British National Corpus*, Oxford University, http://info.ox.ac.uk/bnc/.

Baker, J. K. [1979], "Trainable grammars for speech recognition," *Proceedings of the Spring Conference of the Acoustical Society of America*, Cambridge, MA.

Baum, L. E. & Sell, G. R. [1968], "Growth Transformations for Functions on Manifolds," *Pacific Journal of Mathematics* 27.

Brent, M. R. [1993], "From Grammar to Lexicon: Unsupervised Learning of Lexical Syntax," *Computational Linguistics* 19, 243–262.

Briscoe, T. & Carroll, J. [1996], "Automatic Extraction of Subcategorization from Corpora," *MS* , http://www.cl.cam.ac.uk/users/ejb/.

Charniak, E. [1993], *Statistical Language Learning*, MIT, Cambridge, MA.

——— [1995], "Parsing with Context-free Grammars and Word Statistics," Department of Computer Science, Brown University, Technical Report CS-95-28.

Cover, T. M. & Thomas, J. A. [1991], *Elements of Information Theory*, John Wiley and Sons, Inc., New York.

Dempster, A. P., Laird, N. M. & Rubin, D. B. [1977], "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistics Society* 39, 1–38, Series B.

Ersan, M. & Charniak, E. [1995], "A Statistical Syntactic Disambiguation Program and what it learns," Brown CS Tech Report CS-95-29.

Hornby, A. S. [1985], *Oxford Advanced Learner's Dictionary of Current English*, Oxford University Press, Oxford, 4th Ed..

Jackendoff, R. [1977], *$\overline{X}$ syntax: A study in phrase structure.*, MIT Press, Cambridge, MA.

Katz, S. M. [1980], "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing* 35, 400–401.

Manning, C. [1993], "Automatic acquisition of a large subcategorization dictionary from corpora," *Proceedings of the 31st Annual Meeting of the ACL.*

Neal, R. M. & Hinton, G. E. [1998], "A New View of the EM Algorithm that Justifies Incremental and Other Variants," in *Learning in Graphical Models*, Michael I. Jordan, ed., Kluwer Academic Press.

Ney, H., Essen, U. & Kneser, R. [1994], "On structuring probabilistic dependences in stochastic language modelling," *Computer Speech and Language* 8, 1–38.