Estimators for Stochastic "Unification-Based" Grammars*

Mark Johnson
Cognitive and Linguistic Sciences
Brown University

Stuart Geman
Applied Mathematics
Brown University

Stephen Canon
Cognitive and Linguistic Sciences
Brown University

Zhiyi Chi
Dept. of Statistics
The University of Chicago

Stefan Riezler
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

Abstract

Log-linear models provide a statistically sound framework for Stochastic "Unification-Based" Grammars (SUBGs) and stochastic versions of other kinds of grammars. We describe two computationally-tractable ways of estimating the parameters of such grammars from a training corpus of syntactic analyses, and apply these to estimate a stochastic version of Lexical-Functional Grammar.

1 Introduction

Probabilistic methods have revolutionized computational linguistics. They can provide a systematic treatment of preferences in parsing. Given a suitable estimation procedure, stochastic models can be "tuned" to reflect the properties of a corpus. On the other hand, "Unification-Based" Grammars (UBGs) can express a variety of linguistically-important syntactic and semantic constraints. However, developing Stochastic "Unification-based" Grammars (SUBGs) has not proved as straightforward as might be hoped.

The simple "relative frequency" estimator for PCFGs yields the maximum likelihood parameter estimate, which is to say that it minimizes the Kulback-Liebler divergence between the training and estimated distributions. On the other hand, as Abney (1997) points out, the context-sensitive dependencies that "unification-based" constraints introduce render the relative frequency estimator suboptimal: in general it does not maximize the likelihood and it is inconsistent.

* This research was supported by the National Science Foundation (SBR-9720368), the US Army Research Office (DAAH04-96-BAA5), and Office of Naval Research (N00014-97-1-0249).

Abney (1997) proposes a Markov Random Field or log linear model for SUBGs, and the models described here are instances of Abney's general framework. However, the Monte-Carlo parameter estimation procedure that Abney proposes seems to be computationally impractical for reasonable-sized grammars. Sections 3 and 4 describe two new estimation procedures which are computationally tractable. Section 5 describes an experiment with a small LFG corpus provided to us by Xerox Parc. The log linear framework and the estimation procedures are extremely general, and they apply directly to stochastic versions of HPSG and other theories of grammar.

2 Features in SUBGs

We follow the statistical literature in using the term *feature* to refer to the properties that parameters are associated with (we use the word "attribute" to refer to the attributes or features of a UBG's feature structure). Let Ω be the set of all possible grammatical or well-formed analyses. Each feature f maps a syntactic analysis $\omega \in \Omega$ to a real value $f(\omega)$. The form of a syntactic analysis depends on the underlying linguistic theory. For example, for a PCFG ω would be parse tree, for a LFG ω would be a tuple consisting of (at least) a c-structure, an f-structure and a mapping from c-structure nodes to f-structure elements, and for a Chomskyian transformational grammar ω would be a derivation.

Log-linear models are models in which the log probability is a linear combination of feature values (plus a constant). PCFGs, Gibbs distributions, Maximum-Entropy distributions and Markov Random Fields are all examples of log-linear models. A log-linear model associates each feature f_j with a real-valued parameter θ_j.

A log-linear model with m features is one in which the likelihood $\mathrm{P}(\omega)$ of an analysis ω is:

$$\begin{aligned}
\mathrm{P}_\theta(\omega) &= \frac{1}{Z_\theta} e^{\sum_{j=1,\ldots,m} \theta_j f_j(\omega)} \\
Z_\theta &= \sum_{\omega' \in \Omega} e^{\sum_{j=1,\ldots,m} \theta_j f_j(\omega')}
\end{aligned}$$

While the estimators described below make no assumptions about the range of the f_i, in the models considered here the value of each feature $f_i(\omega)$ is the number of times a particular structural arrangement or configuration occurs in the analysis ω, so $f_i(\omega)$ ranges over the natural numbers.

For example, the features of a PCFG are indexed by productions, i.e., the value $f_i(\omega)$ of feature f_i is the number of times the ith production is used in the derivation ω. This set of features induces a tree-structured dependency graph on the productions which is characteristic of Markov Branching Processes (Pearl, 1988; Frey, 1998). This tree structure has the important consequence that simple "relative-frequencies" yield maximum-likelihood estimates for the θ_i.

Extending a PCFG model by adding additional features not associated with productions will in general add additional dependencies, destroy the tree structure, and substantially complicate maximum likelihood estimation.

This is the situation for a SUBG, even if the features are production occurences. The unification constraints create non-local dependencies among the productions and the dependency graph of a SUBG is usually not a tree. Consequently, maximum likelihood estimation is no longer a simple matter of computing relative frequencies. But the resulting estimation procedures (discussed in detail, shortly), albeit more complicated, have the virtue of applying to essentially arbitrary features—of the production or non-production type. That is, since estimators capable of finding maximum-likelihood parameter estimates for production features in a SUBG will also find maximum-likelihood estimates for non-production features, there is no motivation for restricting features to be of the production type.

Linguistically there is no particular reason for assuming that productions are the best features to use in a stochastic language model. For example, the adjunct attachment ambiguity in (1) results in alternative syntactic structures which use the same productions the same number of times in each derivation, so a model with only production features would necessarily assign them the same likelihood. Thus models that use production features alone predict that there should not be a systematic preference for one of these analyses over the other, contrary to standard psycholinguistic results.

1.a Bill thought Hillary [$_{\mathrm{VP}}$[$_{\mathrm{VP}}$ left] yesterday]

1.b Bill [$_{\mathrm{VP}}$[$_{\mathrm{VP}}$ thought Hillary left] yesterday]

There are many different ways of choosing features for a SUBG, and each of these choices makes an empirical claim about possible distributions of sentences. Specifying the features of a SUBG is as much an empirical matter as specifying the grammar itself. For any given UBG there are a large (usually infinite) number of SUBGs that can be constructed from it, differing only in the features that each SUBG uses.

In addition to production features, the stochastic LFG models evaluated below used the following kinds of features, guided by the principles proposed by Hobbs and Bear (1995). Adjunct and argument features indicate adjunct and argument attachment respectively, and permit the model to capture a general argument attachment preference. In addition, there are specialized adjunct and argument features corresponding to each grammatical function used in LFG (e.g., SUBJ, OBJ, COMP, XCOMP, ADJUNCT, etc.). There are features indicating both high and low attachment (determined by the complexity of the phrase being attached to). Another feature indicates non-right-branching nonterminal nodes. There is a feature for non-parallel coordinate structures (where parallelism is measured in constituent structure terms). Each f-structure attribute-atomic value pair which appears in any feature structure is also used as a feature. We also use a number of features identifying syntactic structures that seem particularly important in these corpora, such as a feature identifying NPs that are dates (it seems that date interpretations of NPs are preferred). We would have liked to have included features concerning specific lexical items (to capture head-to-head dependencies), but we felt that our corpora were so small that the associated parameters could not be accurately estimated.

3 A pseudo-likelihood estimator for log linear models

Suppose $\widetilde{\omega} = \omega_1, \ldots, \omega_n$ is a training corpus of n syntactic analyses. Letting $f_j(\widetilde{\omega}) = \sum_{i=1,\ldots,n} f_j(\omega_i)$, the log likelihood of the corpus $\widetilde{\omega}$ and its derivatives are:

$$\log \mathrm{L}_\theta(\widetilde{\omega}) = \sum_{j=1,\ldots,m} \theta_j f_j(\widetilde{\omega}) - n \log Z_\theta \quad (2)$$

$$\frac{\partial \log \mathrm{L}_\theta(\widetilde{\omega})}{\partial \theta_j} = f_j(\widetilde{\omega}) - n\mathrm{E}_\theta(f_j) \quad (3)$$

where $\mathrm{E}_\theta(f_j)$ is the expected value of f_j under the distribution determined by the parameters θ. The maximum-likelihood estimates are the θ which maximize $\log \mathrm{L}_\theta(\widetilde{\omega})$. The chief difficulty in finding the maximum-likelihood estimates is calculating $\mathrm{E}_\theta(f_j)$, which involves summing over the space of well-formed syntactic structures Ω. There seems to be no analytic or efficient numerical way of doing this for a realistic SUBG.

Abney (1997) proposes a gradient ascent, based upon a Monte Carlo procedure for estimating $\mathrm{E}_\theta(f_j)$. The idea is to generate random samples of feature structures from the distribution $\mathrm{P}_{\hat{\theta}}(\omega)$, where $\hat{\theta}$ is the current parameter estimate, and to use these to estimate $\mathrm{E}_{\hat{\theta}}(f_j)$, and hence the gradient of the likelihood. Samples are generated as follows: Given a SUBG, Abney constructs a covering PCFG based upon the SUBG and $\hat{\theta}$, the current estimate of θ. The derivation trees of the PCFG can be mapped onto a set containing all of the SUBG's syntactic analyses. Monte Carlo samples from the PCFG are comparatively easy to generate, and sample syntactic analyses that do not map to well-formed SUBG syntactic structures are then simply discarded. This generates a stream of syntactic structures, but not distributed according to $\mathrm{P}_{\hat{\theta}}(\omega)$ (distributed instead according to the restriction of the PCFG to the SUBG). Abney proposes using a Metropolis acceptance-rejection method to adjust the distribution of this stream of feature structures to achieve detailed balance, which then produces a stream of feature structures distributed according to $\mathrm{P}_{\hat{\theta}}(\omega)$.

While this scheme is theoretically sound, it would appear to be computationally impractical for realistic SUBGs. Every step of the proposed procedure (corresponding to a single step of gradient ascent) requires a very large number of PCFG samples: samples must be found that correspond to well-formed SUBGs; many such samples are required to bring the Metropolis algorithm to (near) equilibrium; many samples are needed at equilibrium to properly estimate $\mathrm{E}_{\hat{\theta}}(f_j)$.

The idea of a gradient ascent of the likelihood (2) is appealing—a simple calculation reveals that the likelihood is concave and therefore free of local maxima. But the gradient (in particular, $\mathrm{E}_\theta(f_j)$) is intractable. This motivates an alternative strategy involving a data-based estimate of $\mathrm{E}_\theta(f_j)$:

$$\mathrm{E}_\theta(f_j) = \mathrm{E}_\theta(\mathrm{E}_\theta(f_j(\omega)|y(\omega))) \quad (4)$$

$$\approx \frac{1}{n} \sum_{i=1,\ldots,n} \mathrm{E}_\theta(f_j(\omega)|y(\omega) = y_i) \quad (5)$$

where $y(\omega)$ is the yield belonging to the syntactic analysis ω, and $y_i = y(\omega_i)$ is the yield belonging to the i'th sample in the training corpus.

The point is that $\mathrm{E}_\theta(f_j(\omega)|y(\omega) = y_i)$ is generally computable. In fact, if $\Omega(y)$ is the set of well-formed syntactic structures that have yield y (i.e., the set of possible parses of the string y), then

$$\mathrm{E}_\theta(f_j(\omega)|y(\omega) = y_i) = \frac{\sum_{\omega' \in \Omega(y_i)} f_j(\omega') e^{\sum_{k=1,\ldots,m} \theta_k f_k(\omega')}}{\sum_{\omega' \in \Omega(y_i)} e^{\sum_{k=1,\ldots,m} \theta_k f_k(\omega')}}$$

Hence the calculation of the conditional expectations only involves summing over the possible syntactic analyses or parses $\Omega(y_i)$ of the strings in the training corpus. While it is possible to construct UBGs for which the number of possible parses is unmanageably high, for many grammars it is quite manageable to enumerate the set of possible parses and thereby directly evaluate $\mathrm{E}_\theta(f_j(\omega)|y(\omega) = y_i)$.

Therefore, we propose replacing the gradient, (3), by

$$f_j(\widetilde{\omega}) - \sum_{i=1,\ldots,n} \mathrm{E}_\theta(f_j(\omega)|y(\omega) = y_i) \quad (6)$$

and performing a gradient ascent. Of course (6) is no longer the gradient of the likelihood function, but fortunately it is (exactly) the gradient of (the log of) another criterion:

$$\mathrm{PL}_\theta(\widetilde{\omega}) = \prod_{i=1,\ldots,n} \mathrm{P}_\theta(\omega = \omega_i | y(\omega) = y_i) \quad (7)$$

Instead of maximizing the likelihood of the syntactic analyses over the training corpus, we maximize the *conditional* likelihood of these analyses *given the observed yields.* In our experiments, we have used a conjugate-gradient optimization program adapted from the one presented in Press et al. (1992).

Regardless of the pragmatic (computational) motivation, one could perhaps argue that the conditional probabilities $\mathrm{P}_\theta(\omega|y)$ are as useful (if not more useful) as the full probabilities $\mathrm{P}_\theta(\omega)$, at least in those cases for which the ultimate goal is syntactic analysis. Berger et al. (1996) and Jelinek (1997) make this same point and arrive at the same estimator, albeit through a maximum entropy argument.

The problem of estimating parameters for log-linear models is not new. It is especially difficult in cases, such as ours, where a large sample space makes the direct computation of expectations infeasible. Many applications in spatial statistics, involving Markov random fields (MRF), are of this nature as well. In his seminal development of the MRF approach to spatial statistics, Besag introduced a "pseudo-likelihood" estimator to address these difficulties (Besag, 1974; Besag, 1975), and in fact our proposal here is an instance of his method. In general, the likelihood function is replaced by a more manageable product of conditional likelihoods (a *pseudo-likelihood*—hence the designation PL_θ), which is then optimized over the parameter vector, instead of the likelihood itself. In many cases, as in our case here, this substitution side steps much of the computational burden *without sacrificing consistency* (more on this shortly).

What are the asymptotics of optimizing a pseudo-likelihood function? Look first at the likelihood itself. For large n:

$$\begin{aligned}\frac{1}{n}\log \mathrm{L}_\theta(\widetilde{\omega}) &= \frac{1}{n}\log \prod_{i=1,\ldots,n} \mathrm{P}_\theta(\omega_i) \\ &= \frac{1}{n}\sum_{i=1,\ldots,n} \log \mathrm{P}_\theta(\omega_i) \\ &\approx \int \mathrm{P}_{\theta_o}(\omega)\log \mathrm{P}_\theta(\omega)d\omega \quad (8)\end{aligned}$$

where θ_o is the true (and unknown) parameter vector. Up to a constant, (8) is the negative of the Kullback-Leibler divergence between the true and estimated distributions of syntactic analyses. As sample size grows, maximizing likelihood amounts to minimizing divergence. As for pseudo-likelihood:

$$\begin{aligned}\frac{1}{n}\log \mathrm{PL}_\theta(\widetilde{\omega}) &= \frac{1}{n}\log \prod_{i=1,\ldots,n} \mathrm{P}_\theta(\omega = \omega_i|y(\omega) = y_i) \\ &= \frac{1}{n}\sum_{i=1,\ldots,n} \log \mathrm{P}_\theta(\omega = \omega_i|y(\omega) = y_i) \\ &\approx \mathrm{E}_{\theta_o}[\int \mathrm{P}_{\theta_o}(\omega|y)\log \mathrm{P}_\theta(\omega|y)d\omega]\end{aligned}$$

So that maximizing pseudo-likelihood (at large samples) amounts to minimizing the average (over yields) divergence between the true and estimated *conditional distributions of analyses given yields.*

Maximum likelihood estimation is consistent: under broad conditions the sequence of distributions $\mathrm{P}_{\hat{\theta}_n}$, associated with the maximum likelihood estimator for θ_o given the samples $\omega_1, \ldots \omega_n$, converges to P_{θ_o}. Pseudo-likelihood is also consistent, but in the present implementation it is consistent for the conditional distributions $\mathrm{P}_{\theta_o}(\omega|y(\omega))$ and not necessarily for the full distribution P_{θ_o} (see Chi (1998)). It is not hard to see that pseudo-likelihood will not always correctly estimate P_{θ_o}. Suppose there is a feature f_i which depends only on yields: $f_i(\omega) = f_i(y(\omega))$. (Later we will refer to such features as *pseudo-constant.*) In this case, the derivative of $\mathrm{PL}_\theta(\widetilde{\omega})$ with respect to θ_i is zero; $\mathrm{PL}_\theta(\widetilde{\omega})$ contains no information about θ_i. In fact, in this case *any* value of θ_i gives the *same* conditional distribution $\mathrm{P}_\theta(\omega|y(\omega))$; θ_i is irrelevant to the problem of choosing good parses.

Despite the assurance of consistency, pseudo-likelihood estimation is prone to over fitting when a large number of features is matched against a modest-sized training corpus. One particularly troublesome manifestation of over fitting results from the existence of features which, relative to the training set, we might term "pseudo-maximal": Let us say that a feature f is *pseudo-maximal* for a yield y iff

$\forall \omega' \in \Omega(y)\, f(\omega) \geq f(\omega')$ where ω is any correct parse of y, i.e., the feature's value on every correct parse ω of y is greater than or equal to its value on any other parse of y. Pseudo-minimal features are defined similarly. It is easy to see that if f_j is pseudo-maximal on *each sentence* of the training corpus then the parameter assignment $\theta_j = \infty$ maximizes the corpus pseudo-likelihood. (Similarly, the assignment $\theta_j = -\infty$ maximizes pseudo-likelihood if f_j is pseudo-minimal over the training corpus). Such infinite parameter values indicate that the model treats pseudo-maximal features categorically; i.e., any parse with a non-maximal feature value is assigned a zero conditional probability.

Of course, a feature which is pseudo-maximal over the training corpus is not necessarily pseudo-maximal for all yields. This is an instance of over fitting, and it can be addressed, as is customary, by adding a regularization term that promotes small values of θ to the objective function. A common choice is to add a quadratic to the log-likelihood, which corresponds to multiplying the likelihood itself by a normal distribution. In our experiments, we multiplied the pseudo-likelihood by a zero-mean normal in $\theta_1, \ldots \theta_m$, with diagonal covariance, and with standard deviation σ_j for θ_j equal to 7 times the maximum value of f_j found in any parse in the training corpus. (We experimented with other values for σ_j, but the choice seems to have little effect). Thus instead of maximizing the log pseudo-likelihood, we choose $\hat{\theta}$ to maximize

$$\log \mathrm{PL}_\theta(\widetilde{\omega}) - \sum_{j=1,\ldots,m} \frac{\theta_j^2}{2\sigma_j^2} \tag{9}$$

4 A maximum correct estimator for log linear models

The pseudo-likelihood estimator described in the last section finds parameter values which maximize the conditional probabilities of the observed parses (syntactic analyses) given the observed sentences (yields) in the training corpus. One of the empirical evaluation measures we use in the next section measures the number of correct parses selected from the set of all possible parses. This suggests another possible objective function: choose $\hat{\theta}$ to maximize the number $C_\theta(\widetilde{\omega})$ of times the maximum likelihood parse (under θ) is in fact the correct parse, in the training corpus.

$C_\theta(\widetilde{\omega})$ is a highly discontinuous function of θ, and most conventional optimization algorithms perform poorly on it. We had the most success with a slightly modified version of the simulated annealing optimizer described in Press et al. (1992). This procedure is much more computationally intensive than the gradient-based pseudo-likelihood procedure. Its computational difficulty grows (and the quality of solutions degrade) rapidly with the number of features.

5 Empirical evaluation

Ron Kaplan and Hadar Shemtov at Xerox Parc provided us with two LFG parsed corpora. The Verbmobil corpus contains appointment planning dialogs, while the Homecentre corpus is drawn from Xerox printer documentation. Table 1 summarizes the basic properties of these corpora. These corpora contain packed c/f-structure representations (Maxwell III and Kaplan, 1995) of the grammatical parses of each sentence with respect to Lexical-Functional grammars. The corpora also indicate which of these parses is in fact the correct parse (this information was manually entered). Because slightly different grammars were used for each corpus we chose not to combine the two corpora, although we used the set of features described in section 2 for both in the experiments described below. Table 2 describes the properties of the features used for each corpus.

In addition to the two estimators described above we also present results from a baseline estimator in which all parses are treated as equally likely (this corresponds to setting all the parameters θ_j to zero).

We evaluated our estimators using held-out test corpus $\widetilde{\omega}_{\text{test}}$. We used two evaluation measures. In an actual parsing application a SUBG might be used to identify the correct parse from the set of grammatical parses, so our first evaluation measure counts the number $C_{\hat{\theta}}(\widetilde{\omega}_{\text{test}})$ of sentences in the test corpus $\widetilde{\omega}_{\text{test}}$ whose maximum likelihood parse under the estimated model $\hat{\theta}$ is actually the correct parse. If a sentence has l most likely parses (i.e., all l parses have the same conditional probability) and one of these parses is the correct parse, then we score $1/l$ for this sentence.

The second evaluation measure is the pseudo-

	Verbmobil corpus	Homecentre corpus
Number of sentences	540	980
Number of ambiguous sentences	314	481
Number of parses of ambiguous sentences	3245	3169

Table 1: Properties of the two corpora used to evaluate the estimators.

	Verbmobil corpus	Homecentre corpus
Number of features	191	227
Number of rule features	59	57
Number of pseudo-constant features	19	41
Number of pseudo-maximal features	12	4
Number of pseudo-minimal features	8	5

Table 2: Properties of the features used in the stochastic LFG models. The numbers of pseudo-maximal and pseudo-minimal features do not include pseudo-constant features.

likelihood itself, $\mathrm{PL}_{\hat{\theta}}(\tilde{\omega}_{\mathrm{test}})$. The pseudo-likelihood of the test corpus is the likelihood of the correct parses given their yields, so pseudo-likelihood measures how much of the probability mass the model puts onto the correct analyses. This metric seems more relevant to applications where the system needs to estimate how likely it is that the correct analysis lies in a certain set of possible parses; e.g., ambiguity-preserving translation and human-assisted disambiguation. To make the numbers more manageable, we actually present the negative logarithm of the pseudo-likelihood rather than the pseudo-likelihood itself—so smaller is better.

Because of the small size of our corpora we evaluated our estimators using a 10-way cross-validation paradigm. We randomly assigned sentences of each corpus into 10 approximately equal-sized subcorpora, each of which was used in turn as the test corpus. We evaluated on each subcorpus the parameters that were estimated from the 9 remaining subcorpora that served as the training corpus for this run. The evaluation scores from each subcorpus were summed in order to provide the scores presented here.

Table 3 presents the results of the empirical evaluation. The superior performance of both estimators on the Verbmobil corpus probably reflects the fact that the non-rule features were designed to match both the grammar and content of that corpus. The pseudo-likelihood estimator performed better than the correct-parses estimator on both corpora under both evaluation metrics. There seems to be substantial over learning in all these models; we routinely improved performance by discarding features. With a small number of features the correct-parses estimator typically scores better than the pseudo-likelihood estimator on the correct-parses evaluation metric, but the pseudo-likelihood estimator always scores better on the pseudo-likelihood evaluation metric.

6 Conclusion

This paper described a log-linear model for SUBGs and evaluated two estimators for such models. Because estimators that can estimate rule features for SUBGs can also estimate other kinds of features, there is no particular reason to limit attention to rule features in a SUBG. Indeed, the number and choice of features strongly influences the performance of the model. The estimated models are able to identify the correct parse from the set of all possible parses approximately 50% of the time.

We would have liked to introduce features corresponding to dependencies between lexical items. Log-linear models are well-suited for lexical dependencies, but because of the large number of such dependencies substantially larger corpora will probably be needed to estimate such models.[1]

[1] Alternatively, it may be possible to use a simpler non-SUBG model of lexical dependencies estimated from a much larger corpus as the reference distribution with

	Verbmobil corpus		Homecentre corpus	
	$C(\widetilde{\omega}_{\text{test}})$	$-\log \mathrm{PL}(\widetilde{\omega}_{\text{test}})$	$C(\widetilde{\omega}_{\text{test}})$	$-\log \mathrm{PL}(\widetilde{\omega}_{\text{test}})$
Baseline estimator	9.7%	533	15.2%	655
Pseudo-likelihood estimator	58.7%	396	58.8%	583
Correct-parses estimator	53.7%	469	53.2%	604

Table 3: An empirical evaluation of the estimators. $C(\widetilde{\omega}_{\text{test}})$ is the number of maximum likelihood parses of the test corpus that were the correct parses, and $-\log \mathrm{PL}(\widetilde{\omega}_{\text{test}})$ is the negative logarithm of the pseudo-likelihood of the test corpus.

However, there may be applications which can benefit from a model that performs even at this level. For example, in a machine-assisted translation system a model like ours could be used to order possible translations so that more likely alternatives are presented before less likely ones. In the ambiguity-preserving translation framework, a model like this one could be used to choose between sets of analyses whose ambiguities cannot be preserved in translation.

References

Steven P. Abney. 1997. Stochastic Attribute-Value Grammars. *Computational Linguistics*, 23(4):597–617.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

J. Besag. 1974. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society, Series D*, 36:192–236.

J. Besag. 1975. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195.

Zhiyi Chi. 1998. *Probability Models for Complex Systems*. Ph.D. thesis, Brown University.

Brendan J. Frey. 1998. *Graphical Models for Machine Learning and Digital Communication*. The MIT Press, Cambridge, Massachusetts.

Jerry R. Hobbs and John Bear. 1995. Two principles of parse preference. In Antonio Zampolli, Nicoletta Calzolari, and Martha Palmer, editors, *Linguistica Computazionale: Current Issues in Computational Linguistics: In Honour of Don Walker*, pages 503–512. Kluwer.

Frederick Jelinek. 1997. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts.

John T. Maxwell III and Ronald M. Kaplan. 1995. A method for disjunctive constraint satisfaction. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell III, and Annie Zaenen, editors, *Formal Issues in Lexical-Functional Grammar*, number 47 in CSLI Lecture Notes Series, chapter 14, pages 381–481. CSLI Publications.

Judea Pearl. 1988. *Probabalistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipies in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 2nd edition.

respect to which the SUBG model is defined, as described in Jelinek (1997).