Fast, Piecewise Training for Discriminative Finite-state and Parsing Models

Charles Sutton and **Andrew McCallum**
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003 USA
{casutton,mccallum}@cs.umass.edu

Abstract

Discriminitive models for sequences and trees—such as linear-chain conditional random fields (CRFs) and max-margin parsing—have shown great promise because they combine the ability to incorporate arbitrary input features and the benefits of principled global inference over their structured outputs. However, since parameter estimation in these models involves repeatedly performing this global inference, training can be very slow. We present *piecewise training*, a new training method that combines the speed of local training with the accuracy of global training by incorporating a limited amount of global information derived from previous errors of the model. On named-entity and part-of-speech data, we show that our new method not only trains in less than one-fifth the time of a CRF and yields improved accuracy over the MEMM, but surprisingly also provides a statistically-significant gain in accuracy over the CRF. Also, we present preliminary results showing a potential application to efficient training of discriminative parsers.

1 Introduction

Conditionally-trained models have enjoyed popularity for a wide variety of tasks in NLP, including document classification (Taskar et al., 2002; Nigam et al., 1999), part-of-speech tagging (Ratnaparkhi, 1996; Toutanova et al., 2003), chunking (Sha and Pereira, 2003), named-entity recognition (Florian et al., 2004), and information extraction (McCallum et al., 2000; Pinto et al., 2003). Their popularity stems from the flexibility they afford in designing rich features to best suit particular tasks.

Of special recent interest has been discriminative parsing (Taskar et al., 2004; Clark and Curran, 2004; Collins and Roark, 2004). Discriminative parsing models have been interesting, because the rich features sets they afford can allow higher better accuracy without explicitly needing the intricate smoothing required by state-of-the-art generative parsing models (Charniak, 2000; Collins, 2000).

These benefits come at a cost, however. Although conditional training for unstructured models is relatively efficient (it is simply the maximum-entropy classifier), if the model predicts richly structured outputs, such as sequences or parse trees, then discriminative training can become extremely expensive. This is because learning with structured outputs requires repeated inference over the training set: to compute model expectations in CRF-style training, or to find a high-probability configuration in perceptron training.

Discriminative parsers, for instance, can be fantastically expensive to train, because they require repeatedly parsing the many sentences in the training set. For example, Clark and Curran (2004) report training their model on 45 machines in parallel. And a max-margin parsing model can take over 3 months to train on the entire Penn treebank (Ben Taskar, per-

sonal communication).

Even in simpler models, such as for sequences, CRF training can be slow if there are many states. Training a CRF part-of-speech tagger on the Sections 2-21 of the Penn treebank, for example, can take weeks because of expense of running forward-backward with 45 part-of-speech tags.

Conditional training methods like CRF and perceptron training can be expensive because they are *global*, that is, they require repeatedly performing inference over entire structured training examples. An alternative is to define a model that is locally normalized, for example, in terms of distributions over successor states or individual expansions of a nonterminal. Locally-normalized models correspond to learning probabilistic classifiers for individual decisions in the model. Thus training can be much faster because it does not require global inference. Unfortunately, globally-trained models usually perform better. One reason for this is that when an earlier classifier makes a mistake, a later classifier has local information that indicates against it. But even if global inference is performed at test time, because training is local, the later classifiers never learn to vote against earlier mistakes.

These phenomena appear even in models as simple as a linear chain. A CRF is a globally-normalized sequence model, in which parameter estimation involves repeatedly running forward-backward over the training set, which can be expensive if the state space is large. Alternatively, maximum-entropy markov models (MEMMs) are locally-normalized conditional sequence models, in which the parameter estimation for each next-state classifier can be performed separately, and forward-backward is not required. But despite their fast training, MEMMs are known to suffer from several pathologies, such as label bias (Lafferty et al., 2001) and observation bias (Klein and Manning, 2002). For these reasons, across a broad range of NLP tasks, MEMMs consistently perform worse than CRFs.

We propose a new piecewise training procedure for locally-normalized models that allows limited interaction between the local classifiers. This combines the efficiency of MEMM-style training with the accuracy of CRF-style training. The basic idea is to train the the next-state classifier to recognize and vote against errors in the previous state. We do this by augmenting the local training sets of an MEMM with noisy instances where the source state is actually in error; for these new instances, instead of predicting a next-state, the classifier is trained to predict a new "none-of-the-above" (NOTA) label, thereby learning to vote against the previous incorrect decision. In two different NLP tasks, we show that piecewise training not only is faster than CRF training but also, amazingly, has higher accuracy. We hypothesize that the CRF's capacity to trade off weights across sequence positions leads to greater capacity to overfit.

Although we explore piecewise training in the case of linear models, it can be applied to general probability distributions, as long as they can be divided into locally-normalized pieces. To demonstrate this, we also present early, preliminary results applying piecewise training to a discriminative parsing model. With piecewise training, we show a small improvement over an MEMM-style discriminative parsing model.[1] Unlike other discriminative parsing approaches, our piecewise procedure requires parsing the training set only once.

2 Conditional Linear-Chain Models

In this section, we review two standard models for sequence labeling, the *maximum-entropy Markov model* (MEMM) (McCallum et al., 2000) and the *conditional random fields* (CRF) (Lafferty et al., 2001). Both CRFs and MEMMs are methods for conditionally training weighted finite state transducers, so that given a sentence $\mathbf{x}$, both assign a probability $p(\mathbf{y}|\mathbf{x})$ to a sequence of labels $\mathbf{y}$. Both models are defined in terms of a set of features $\{f_k(y_t, y_{t-1}, \mathbf{x}, t)\}$ on state transitions, each of which has an associated real-valued weight λ_k.

MEMMs learn a next-state classifier for each state in the FSM. In an MEMM, the next-state classifier trained by maximum entropy, so that the model is given by:

$$p(y_{t-1}|y_t, \mathbf{x}) = \frac{\exp\left(\sum_k \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}_t)\right)}{Z(\mathbf{x}_t, y_t)} \quad (1)$$

[1] In fact, the MEMM-style parsing model, which we term an MECFG, seems unexplored in the literature, and may be interesting in its own right.

where each next-state distribution is locally normalized by

$$Z(\mathbf{x}_t, y_t) = \sum_{y'} \exp\left(\sum_k \lambda_k f_k(y', y_t, \mathbf{x}_t)\right) \quad (2)$$

Several pathologies have been observed with MEMMs. One such pathology is *label bias* (Bottou, 1991; Lafferty et al., 2001), in which the MEMM tends to favor states with low-entropy next-state distributions. In the extreme, states with only one successor ignore their observations completely.

Now, if the Markov assumptions of the MEMM were accurate, ignoring the observation would be correct; label bias arises because a later observation influences an earlier transition. In fact, in carefully designed experiments with synthetic data it has been reported (Lafferty et al., 2001) that CRFs are more robust to violations of their independence assumptions than MEMMs.

Also, Klein and Manning (Klein and Manning, 2002) report the phenomenon of *observation bias*, in which in some data sets, instead of observations being incorrectly ignored, state transitions are not given enough weight, even when they are highly predictive. Now, this phenomenon has been reported in part-of-speech tagging, where the observations are known to provide much more information than the state transitions, so it is not clear to what extent this phenomenon occurs across tasks.

Because of these pathologies, MEMMs have been shown to perform worse than CRFs across NLP data sets, including noun-phrase chunking (Sha and Pereira, 2003), part-of-speech tagging, and named-entity recognition (Section 4).

The CRF remedies these limitations of the MEMM. A linear-chain CRF defines a global distribution over label sequences $\mathbf{y}$ as

$$p(\mathbf{y}|\mathbf{x}) = \frac{\prod_t \exp\left(\sum_k \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}_t)\right)}{Z(\mathbf{x})}, \quad (3)$$

where $Z(\mathbf{x})$ is a normalization constant given by

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_t \exp\left(\sum_k \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}_t)\right). \quad (4)$$

The only difference between the CRF model and the MEMM model, then, is that a CRF is globally normalized over all possible label sequences, while in an MEMM, each transition distribution is locally normalized.

Conditional random fields are usually trained by maximum likelihood. The partial derivative of the likelihood with respect to a weight λ_k is

$$\frac{\partial \mathcal{L}}{\partial \lambda_k} = \sum_t f_k(y_t, y_{t-1}, \mathbf{x}_t) - \sum_t \sum_{y,y'} f_k(y, y', \mathbf{x}) p(y, y'|\mathbf{x}) \quad (5)$$

Computing the gradient thus requires computing $p(y_t, y_{t-1}|\mathbf{x})$, which can be computed by the forward-backward algorithm. This explains the difference in training time between CRFs and MEMMs. Computing the MEMM gradient requires local computations that are linear in the number of FSM states. The CRF gradient, on the other hand, requires running forward-backward, which is quadratic in the number of states. And forward-backward must be called once for each training instance at each iteration of a numerical optimization procedure. This means that CRF training as a whole can require hundreds of thousands of calls to forward-backward for large NLP data sets.

3 Piecewise Training with Limited Interactions

In this section, we introduce *piecewise training*, a method for local training that avoids the pathologies of MEMMs. For concreteness, we present our method for the case of linear chains, although it is more general.

In MEMMs, training by conditioning only on the true values of the previous state can be problematic. When global inference at test time estimates high probability for an incorrect state at the previous time step, the next-state classifiers are evaluated on inputs they may never have seen at training time, resulting in unpredictable scores.

We propose avoiding this problem by augmenting the objective function of the MEMM. For each local classifier we introduce an additional value termed "none of the above," or NOTA (e.g., an additional, imaginary part-of-speech label). Traditional MEMM training would create local training sets for

each source state, assigning training data to a particular next-state classifier using the true labels in the training data. By contrast, we also assign training data to a source state even when the true source state from the original training sequence does not match; the correct predicted value in this case is NOTA.

Inference at test time is performed with standard MEMM local inference, with the NOTA state and all parameters associated with NOTA outcomes removed. Accordingly, training is performed such that all parameters associated with NOTAvalues are constrained to be zero. Thus the only way for NOTA to be correctly predicted is by reducing the strength of the parameters associated with other outcomes, given the current observations. Once NOTA is removed, the next-state distribution is nearly uniform, which in an MEMM is the next-state distribution that most equally "votes against" all possible outcomes.

Formally, the objective function used in a piecewise-trained linear-chain model with parameters Λ and training data $\mathcal{D} = \{\langle \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \rangle\}$ is

$$\mathcal{O}_{PT}(\Lambda, \mathcal{D}) = \log \prod_i^{|\mathcal{D}|} \prod_t \tilde{p}_\Lambda(y_t^{(i)} | y_{t-1}^{(i)}, \mathbf{x}_t^{(i)}) \prod_{y \neq y_{t-1}^{(i)}} \tilde{p}_\Lambda(\text{NOTA} | y, \mathbf{x}_t^{(i)}). \quad (6)$$

An interesting insight comes from expanding this to show the normalization function and the product of potential functions. If we define a potential function as

$$\phi(y_t, y_{t-1}, \mathbf{x}_t) = \exp\left(\sum_k \lambda_k f_k((y_t, y_{t-1}, \mathbf{x}_t)\right) \quad (7)$$

then we see

$$\begin{aligned}\mathcal{O}_{PT}(\Lambda, \mathcal{D}) &= \log \prod_i^{|\mathcal{D}|} \prod_t \frac{\phi(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)})}{1 + \sum_y \phi(y, y_{t-1}^{(i)}, x_t^{(i)})} \prod_{y' \neq y_{t-1}^{(i)}} \frac{1}{1 + \sum_y \phi(y, y', x_t^{(i)})} \\ &= \log \prod_i^{|\mathcal{D}|} \prod_t \frac{\phi(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)})}{\prod_{y'} \left(1 + \sum_y \phi(y, y', x_t^{(i)})\right)},\end{aligned}$$

where the sum over y does not include NOTA (since they are captured with the included 1's). This corresponds to approximating the normalization function $Z(\Lambda, \mathbf{x})$ with

$$Z_{PT}(\Lambda, \mathbf{x}^{(i)}) = \prod_t \prod_{y'} \left(1 + \sum_y \phi(y, y', x_t^{(i)})\right).$$

Tom Minka (personal communication) has pointed out that this seems to be a novel approximation to the partition function, and that a somewhat similar approximation is produced in the very beginning of belief propagation, when all messages are equal to 1, and the partition function is estimated by

$$Z_{BP}(\Lambda, \mathbf{x}^{(i)}) \propto \prod_t \sum_{y'} \sum_y \phi(y, y', x_t^{(i)}).$$

The training data for the NOTA outcome, $y \neq y_{t-1}^{(i)}$, may be exhaustive, or randomly sampled, or chosen to include only those cases in which incorrectly had high marginal probability by joint inference with a previous parameter setting. A method similar to this last option has been previously used to successfully incorporate not all but some of the most important "unsupported features" in linear-chain CRFs (McCallum, personal communication).

Furthermore, we can calibrate the magnitude of the parameters Λ_s across each subset s, by learning a per-subset multiplicative factor, $\alpha_s \Lambda_s$. Although this factor is learned via traditional global inference, its impact on training time is limited because it has such low dimensionality that optimization typically requires only a few gradient steps.

Essentially NOTA outcomes allow limited communication between locally-normalized subsets, by allowing them to assign uninformative distributions to incorrect variable assignments of conditioned variables.

4 Experiments on Sequential Tasks

Although piecewise training was motivated by the need to train large structure models such as parse trees, we show here that piecewise training can be beneficial even in graphical models as simple as a linear-chain.

We present results on two tasks: part-of-speech tagging and named-entity recognition. First, for

	Named entity			POS tagging		
	Training F1	Testing F1	Training Time	Training Accuracy	Testing Accuracy	Training Time
MEMM	99.89%	88.90	1 hr	99.1%	88.1%	2 hr, 8 min
CRF	99.95%	89.87	9 hr	99.8%	88.1%	14 hr
CRF-PT	99.82%	90.47	5 hr, 35 min	99.08%	88.8%	2 hr, 30 min

Table 1: Results on named-entity recognition and part-of-speech tagging. CRFs trained in pieces (CRF-PT) statistically-significantly outperform both regular MEMMs and CRFs. A small subset of the treebank was used for the POS results, which explains the low baseline performance.

named-entity recognition, we use the CoNLL 2003 English data set, consisting of 14,987 newswire sentences annotated with names of people, organizations, locations, and miscellaneous entities. We test on the standard development set of 3,466 sentences. Evaluation is done using precision and recall on the extracted chunks, and we report $F_1 = 2PR/P + R$.

Results are shown in Table 1. We compare a CRF, an MEMM, and a piecewise-trained CRF (CRF-PT) with exhaustively-added NOTA instances. Consistent with previous work, the CRF performs better than the MEMM. But with the addition of NOTA instances, the CRF-PT performs better than the standard MEMM and, amazingly, also better than the CRF. It appears that CRF-PT is overfitting less than the CRF, since CRF-PT has lower training accuracy despite its higher testing accuracy. All of the pairwise differences in table 1 are significant by McNemar's test on the per-sentence labeling disagreements ($p < 0.001$).

Second, in previous work (Lafferty et al., 2001), CRFs were shown to outperform MEMMs on part-of-speech tagging. Here we test whether training in pieces addresses the previously-observed problems with local normalization on a POS data set. For these preliminary experiments, we used a small subset comprising 1,154 sentences, randomly sampled from sections 0–18 of the Penn Treebank WSJ corpus. We evaluated on all 5,527 sentences of of sections 20 and 21. The Treebank tag set contains 45 tags.

In this experiment, we achieved better performance by including only a few NOTA interactions. In particular, after twenty iterations of training, we added a NOTA term of the form $p(\text{NOTA}|y_t)$ for all incorrect y_t in the training set that the model assigned probability greater than 0.2.

On this small training set, the MEMM and the CRF had identical performance. The training set is so small that the CRF's greater capacity to overfit negates its advantage in avoiding label bias. When trained on larger subsets of the treebank, the CRF performs better than the the MEMM, consistent with previous results. Still, the piecewise-trained CRF-PT achieves significantly better performance than both the CRF and the MEMM. This difference is significant by a paired t-test on the number of incorrect tags per sentence ($p < 0.001$).

In both data sets, we found that for piecewise training using local MEMM-style normalization at test time performed better than CRF-style global testing. This is not surprising, because one might expect that the weights from each of the separately-trained pieces would have different scale. For the NER results that we report, we globally train a per-state scaling factor, as mentioned in the previous section. For the POS results in Table 1, however, we used locally-normalized MEMM testing.

5 Discriminative Parsing

The real promise of piecewise training techniques is for richly-structured models that are simply too complex to train globally. In this section, we present early results which suggest that piecewise training is also a promising approach for discriminative parsing, which can take weeks or months to train globally.

Our aim in these early experiments is not to beat generative models (by performing extensive feature engineering), but to show that piecewise training improves over a plain locally-normalized MEMM analogue to parsing. We indeed find a small improve-

	Number NOTA Instances	Test P (len<= 15)	Test R (len<= 15)
GENERATIVE	N/A	81.7	83.1
MECFG	0	81.4	82.7
PT-CFG	1 564 827	81.7	83.2

Table 2: Early results from a piecewise-trained discriminative parsing model. As more NOTA instances are added, the performance of the model improves, evntually equalling the generative baseline.

ment, suggestive of good future work in this area.

Although previous locally-normalized parsers (Ratnaparkhi, 1999) have normalized over shift-reduce decisions, a locally-normalized model can be defined from a CFG more directly. Just as a MEMM locally normalizes over all next-states from a source state, in a CFG one can normalize over all expansions of a given chart edge. This yields a conditional model from a PCFG exactly as an MEMM does from an HMM.

More formally, let T be a parse tree for a sentence $\mathbf{x}$. We assume that we have a CFG G in Chomsky normal form, and set of features $f_k(A, BC, \mathbf{x}, i, j)$, where $A \rightarrow BC$ is a rule in the CFG, and the indices i and j are the boundaries in $\mathbf{x}$ of the subtree headed by A. This means essentially that all features must computable given a single traversal in a chart.

To define the probability of a parse tree in this model, we first define the probability of a single expansion $A \rightarrow BC$ occuring when A spans the sequence $x_i \ldots x_j$:

$$p(BC|A, x_i \ldots x_j) = \frac{\phi(A, BC, \mathbf{x}, i, j)}{Z(A, \mathbf{x}, i, j))}$$

$$Z(A, \mathbf{x}, i, j) = \sum_{B'C' : A' \rightarrow B'C'} \phi(A, BC, \mathbf{x}, i, j)$$

$$\phi(A, BC, \mathbf{x}, i, j) = \exp\left(\sum_k \lambda_k f_k(A, BC, \mathbf{x}, i, j)\right),$$

and then the probability of a tree T is simply the product of all expansions that occur in it. We are not aware of this model in the literature, so we call it a *maximum-entropy context-free grammar* (MECFG).

Maximum-likelihood training for MECFGs can be accomplished by numerical optimization, as is standard for maximum-entropy models. Thus, although MECFG training is more expensive than PCFG training, it still requires no parsing at training time.

We report early results from training these models on the Penn Wall St. Journal Treebank. In the results here, we restrict the training and test sets sentences of length ≤ 15 words. As usual, we report labeled precision and recall. We train on sections 2-21 (9753 sentences), and use section 22 as our test set (421 sentences).

We use the CFG structure from the unlexicalized PCFG of Klein and Manning (2003).[2] However, in these results, our tagging model is not very sophisticated, and our handling of unknown words is very simple. This explains why our generative baseline has an F_1 of 82.4, much lower than the 88 F1 that one can get with this PCFG structure on this subset.

Our features are lexical features of the span boundaries, similar to Taskar et al. (Taskar et al., 2004). Specifically, we used the first and last words of the span, conjoined with the span length if it is less than 3.

To apply piecewise training to this model, recall that a NOTA instance corresponds to an incorrect decision that could potentially occur during global inference at test time. While in an MEMM, this corresponds to an incorrect source state, in a CFG this corresponds to an incorrect chart edge (X, i, j). A chart edge could be incorrect for two reasons: the nonterminal X can be incorrect, and the span (i, j) might not correspond to a true bracketing. We selectively generate NOTA instances for both kinds of errors, based on mistakes made by a partially-trained model. Specifically, after 10 iterations of MECFG training, we parse the training set and add NOTA

[2] In fact, our implementation uses transformed trees printed directly from Dan Klein's code, which is available at `http://www-nlp.stanford.edu/software/lex-parser.shtml`.

instances that correspond to incorrect chart edges whose scores are sufficiently high. For each incorrect chart edge, we compare its score to the best-scoring chart edge of the same length; if the log ratio of scores is greater than a threshold δ (here, we use $\delta = 5$), then the incorrect chart edge is included as a NOTA instance.

Results from comparing MECFG training, piecewise training, and a generative baseline are given in Table 2. We expected the MECFG to perform poorly, perhaps suffering even more severely from the same pathologies that affect MEMMs. To our surprise, this was not the case; the MECFG comes within 0.4 F_1 of the generative baseline. Even so, piecewise training still provides a small, suggestive improvement, essentially equalling the generative baseline in performance. More complex discriminative features, such as scores from a generative model (Collins and Roark, 2004) and richer lexical features, can help performance further.

We have shown that with fairly basic lexical features, piecewise training can equal the performance of a generative baseline without the vast training time required by a global discriminative model. To get an idea of the cost of training a global model on this subset, our parser takes about 40 minutes to parse the entire training set. Suppose that global training takes 100 iterations to converge (CRF-style training might use many more iterations than this; perceptron training may use less.) This yields a back-of-the-envelope estimate of 3 days to train. By contrast, piecewise training took only 4.6 hr. In addition, we do not necessarily need to parse the entire training set to generate NOTA instances; parsing a sample of the training set may work well for large data sets.

6 Related Work

There are several examples in the literature of undirected models trained in locally-normalized pieces. Pseudolikelihood (Besag, 1975) is a well-known method for training a globally-normalized model using local distributions. In pseudolikelihood, parameters are trained to maximize the likelihood of each predicted variable, conditioned on the true values of the neighboring variables. The MEMM training objective is actually very similar to the pseudolikelihood objective, except that in the MEMM objective, the local term for each node is conditioned only on the previous node, not on both neighbors as in pseudolikeliood. It would be interesting to see whether the NOTAtechnique can be used to improve the performance of pseudolikelihood training as well. The MEMM objective has also been used by others (Punyakanok and Roth, 2001; Klein et al., 2003).

Pseudolikelihood has had some success in applications. For example, Toutanova et al. (2003) achieve state-of-the-art performance on part-of-speech tagging using a cyclic dependency network trained using pseudolikelihood. Also, pseudolikelihood has been used for grid-shaped CRFs in computer vision (Kumar and Hebert, 2003).

Roth (2002) has advocated training disjoint classifiers, and then performing joint inference at test time in an approach he terms "training with classifiers."

Kakade, Teh, and Roweis (2002) show that label bias in MEMMs can be somewhat ameliorated by training on the marginal probability of single labels. With this training objective, MEMMs actually perform better on token accuracy than CRFs on an extraction data set. To compute the marginal likelihood, however, requires forward-backward, and therefore is just as computationally intensive as global CRF training.

7 Conclusion

We present a new method for efficient piecewise of large, richly structure probabilistic models. Using so-called "none-of-the-above" instances we allow some global interactions across the model without requiring full inference at training time. On two sequence-labeling NLP tasks, we show that NOTA training preserves the efficiency of MEMM training, while surprisingly achieving significantly better accuracy than a CRF. We also present early results showing that piecewise training is a promising approach for discriminative parsing. We introduce a *maximum-entropy context free grammar*, a locally-normalized parsing model analogous to MEMM, which we believe to be novel. On a subset of the Penn treebank, we show that MECFGs perform surprisingly well, only slightly below an (admittedly low) generative baseline. We present early results suggesting that piecewise training can potentially

improve performance over a MECFG.

In future work, we will improve the parsing results, by using better tagging models in the baseline, and by making better use of discriminative features. Also, we are interested in broader applications of piecewise training to large structured models, such as arise in information extraction and data mining.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant numbers IIS-0326249 and IIS-0427594. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

Julian Besag. 1975. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195.

Léon Bottou. 1991. *Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole*. Ph.D. thesis, Université de Paris XI, Orsay, France.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 132–139. Morgan Kaufmann Publishers Inc.

Stephen Clark and James R. Curran. 2004. Parsing the wsj using ccg and log-linear models. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 103–110, Barcelona, Spain, July.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 111–118, Barcelona, Spain, July.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA.

Radu Florian, H. Hassan, A. Ittycheriah, Hongyan Jing, Nanda Kambhatla, X. Luo, Nicolas Nicolov, Salim Roukos, and Tong Zhang. 2004. A statistical model for multilingual entity detection and tracking. In *In* HLT/NAACL 2004.

Sham Kakade, Yee Whye Teh, and Sam Roweis. 2002. An alternative objective function for markovian fields. In *In Proceedings of the Nineteenth International Conference on Machine Learning*.

Dan Klein and Christopher D. Manning. 2002. Conditional structure versus conditional estimation in nlp models. In *EMNLP*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D. Manning. 2003. Named entity recognition with character-level models. In *Proceedings the Seventh Conference on Natural Language Learning*, pages 180–183.

Sanjiv Kumar and Martial Hebert. 2003. Discriminative fields for modeling spatial dependencies in natural images. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA.

Kamal Nigam, John Lafferty, and Andrew McCallum. 1999. Using maximum entropy for text classification. In *IJCAI'99 Workshop on Information Filtering*.

David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft. 2003. Table extraction using conditional random fields. In *Proceedings of the ACM SIGIR*.

V. Punyakanok and D. Roth. 2001. The use of classifiers in sequential inference. In *NIPS 13*.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. of the 1996 Conference on Empirical Methods in Natural Language Proceeding (EMNLP 1996)*.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34:151–175.

Dan Roth. 2002. Reasoning with classifiers. In *Proc. of European Conference on Machine Learning*.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL 2003*. Association for Computational Linguistics.

Ben Taskar, Pieter Abbeel, and Daphne Koller. 2002. Discriminative probabilistic models for relational data. In *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*.

Ben Taskar, Dan Klein, Michael Collins, Daphne Koller, and Chris Manning. 2004. Max-margin parsing. In *Empirical Methods in Natural Language Processing (EMNLP04)*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL 2003*.