Conditional Structure versus Conditional Estimation in NLP Models

Dan Klein and **Christopher D. Manning**
Computer Science Department
Stanford University
Stanford, CA 94305-9040
{klein, manning}@cs.stanford.edu

Abstract

This paper separates conditional *parameter estimation*, which consistently raises test set accuracy on statistical NLP tasks, from conditional *model structures*, such as the conditional Markov model used for maximum-entropy tagging, which tend to lower accuracy. Error analysis on part-of-speech tagging shows that the actual tagging errors made by the conditionally structured model derive not only from label bias, but also from other ways in which the independence assumptions of the conditional model structure are unsuited to linguistic sequences. The paper presents new word-sense disambiguation and POS tagging experiments, and integrates apparently conflicting reports from other recent work.

1 Introduction

The success and widespread adoption of probabilistic models in NLP has led to numerous variant methods for any given task, and it can be difficult to tell what aspects of a system have led to its relative successes or failures. As an example, maximum entropy taggers have achieved very good performance (Ratnaparkhi, 1998; Toutanova and Manning, 2000; Lafferty et al., 2001), but almost identical performance has also come from finely tuned HMM models (Brants, 2000; Thede and Harper, 1999). Are any performance gains due to the sequence model used, the maximum entropy approach to parameter estimation, or the features employed by the system?

Recent experiments have given conflicting recommendations. Johnson (2001) finds that a conditionally trained PCFG marginally outperforms a standard jointly trained PCFG, but that a conditional shift-reduce model performs worse than a joint formulation. Lafferty et al. (2001) suggest on abstract grounds that conditional models will suffer from a phenomenon called *label bias* (Bottou, 1991) – see section 3 – but is this a significant effect for real NLP problems?

We suggest that the results in the literature, along with the new results we present in this work, can be explained by the following generalizations:

- The ability to include better features in a well-founded fashion leads to better performance.
- For fixed features, assumptions implicit in the model structure have a large impact on errors.
- Maximizing the objective being evaluated has a reliably positive, but often small, effect.

It is especially important to study these issues using NLP data sets: NLP tasks are marked by their complexity and sparsity, and, as we show, conclusions imported from the machine-learning literature do not always hold in these characteristic contexts.

In previous work, the structure of a model and the method of parameter estimation were often both changed simultaneously (for reasons of naturalness or computational ease), but in this paper we seek to tease apart the separate effects of these two factors. In section 2, we take the Naive-Bayes model, applied to word-sense disambiguation (WSD), and train it to maximize various objective functions. Our experiments reaffirm that discriminative objectives like conditional likelihood improve test-set accuracy. In section 3, we examine two different model structures for part-of-speech (POS) tagging. There, we analyze how assumptions latent in conditional structures lower tagging accuracy and produce strange qualitative behaviors. Finally, we discuss related recent findings by other researchers.

2 Objective Functions: Naive-Bayes

For bag-of-words WSD, we have a corpus D of labeled examples $(s, \mathbf{o})$. Each $\mathbf{o} = \langle o_i \rangle$ is a list of context words, and the corresponding s is the correct sense of a fixed target word occuring in that context. A particular model for this task is the familiar multinomial *Naive-Bayes* (NB) model (Gale et al., 1992; McCallum and Nigam, 1998), where we assume conditional independence between each of the o_i. This NB model gives a joint distribution over the s and $\langle o_i \rangle$ variables:

$$P(s, \mathbf{o}) = P(s) \prod_i P(o_i|s)$$

It also implicitly makes conditional predictions:

$$P(s|\mathbf{o}) = P(s, \mathbf{o}) / \sum_{s'} P(s', \mathbf{o})$$

In NLP, NB models are typically used in this latter way to make conditional decisions, such as chosing the most likely word sense.[1]

The parameters $\Theta = \langle \theta_s; \theta_{o|s} \rangle$ for this model are the sense priors $P(s)$ and the sense-conditional word distributions $P(o|s)$. These are typically set using (smoothed) *relative frequency estimators* (RFEs):

$$\begin{aligned} \theta_s &= P(s) = \text{count}(s)/|D| \\ \theta_{o|s} &= P(o|s) = \text{count}(s, o) / \sum_{o'} \text{count}(s, o') \end{aligned}$$

These intuitive relative frequency estimators are the estimates for Θ which maximize the *joint likelihood* (JL) of D according to the NB model:

$$JL(\Theta, D) = \prod_{(s,\mathbf{o}) \in D} P(s, \mathbf{o})$$

A NB model which has been trained to maximize JL will be referred to as NB-JL. It is worth emphasizing that, in NLP applications, the model is typically trained jointly, then used for its $P(s|\mathbf{o})$ predictions.

We can set the parameters in other ways, without changing our model. If we are doing classification, we may not care about JL. Rather, we will want to minimize whatever kinds of errors we get charged for. The JL objective is the evaluation criterion for language modeling, but a decision process' evaluation is more naturally phrased in terms of $P(s|\mathbf{o})$. If we want to maximize the probability assigned to the correct labeling of the corpus, the appropriate objective is *conditional likelihood* (CL):

$$CL(\Theta, D) = \prod_{(s,\mathbf{o}) \in D} P(s|\mathbf{o})$$

This focuses on the sense predictions, not the words, which is what we cared about in the first place.

Figure 1 shows an example of the trade-offs between JL and CL. Assume there are two classes (1 and 2), two words (a and b), and only 2-word contexts. Assume the actual distribution (training and test) is 3 each of (1, *ab*) and (1, *ba*) and one (2, *aa*) for 7 samples. Then, as shown in figure 1, the JL-maximizing NB model has priors of 6/7 and 1/7, like the data. The actual (joint) distribution is not in the family of NB models, and so it cannot be learned perfectly. Still, the NB-JL assigns reasonable probabilities to all occurring events. However, its priors cause it to incorrectly predict that *aa* belongs to class 1. On the other hand, maximizing CL will push the prior for sense 1 arbitrarily close to zero. As a result, its conditional predictions become more accurate at the cost of its joint prediction. NB-CL joint prediction assigns vanishing mass to events other than (2, *aa*), and so its joint likelihood score gets arbitrarily bad.

			$P(s, \mathbf{o})$			$P(s\|\mathbf{o})$			Correct?	
s	$\mathbf{o}$	Counts	Actual	NB-JL	NB-CL	Actual	NB-JL	NB-CL	NB-JL	NB-CL
1	aa	0	0	3/14	$\epsilon/4$	0	3/5	$\epsilon/4$		
1	ab	3	3/7	3/14	$\epsilon/4$	1	1	1	+	+
1	ba	3	3/7	3/14	$\epsilon/4$	1	1	1	+	+
1	bb	0	0	3/14	$\epsilon/4$	0	1	1		
2	aa	1	1/7	1/7	$1-\epsilon$	1	2/5	$1-\epsilon/4$	-	+
2	ab	0	0	0	0	0	0	0		
2	ba	0	0	0	0	0	0	0		
2	bb	0	0	0	0	0	0	0		
Limit log prod.			-0.44	-0.69	$-\infty$	0.00	-0.05	0.00		
						Accuracy			6/7	7/7

Model	$P(1)$	$P(2)$	$P(a\|1)$	$P(b\|1)$	$P(a\|2)$	$P(b\|2)$
NB-JL	6/7	1/7	1/2	1/2	1	0
NB-CL	ϵ	$1-\epsilon$	1/2	1/2	1	0

Figure 1: Example of joint vs. conditional estimation.

There are other objectives (or *loss functions*). In the SENSEVAL competition (Kilgarriff, 1998), we guess sense distributions, and our score is the sum of the masses assigned to the correct senses. This objective is the *sum of conditional likelihoods* (SCL):

$$SCL(\Theta, D) = \sum_{(s,\mathbf{o}) \in D} P(s|\mathbf{o})$$

SCL is less appropriate that CL when the model is used as a step in a probabilistic process, rather than in isolation. CL is more appropriate for filter processes, because it highly punishes assigning zero or near-zero probabilities to observed outcomes.

If we choose single senses and receive a score of either 1 or 0 on an instance, then we have *0/1-loss* (Friedman, 1997). This gives the "number correct" and so we refer to the corresponding objective as *accuracy* (Acc):

$$Acc(\Theta, D) = \sum_{(s,\mathbf{o}) \in D} \delta(s = \arg\max_{s'} P(s'|\mathbf{o}))$$

In the following experiments, we illustrate that, for a fixed model structure, it is advantageous to maximize objective functions which are similar to the evaluation criteria. Although in principle we can optimize any of the objectives above, in practice some are harder to optimize than others. As stated above, JL is trivial to maximize with a NB model. CL and

[1] A possible use for the joint predictions would be a topic-conditional unigram language model.

SCL, since they are continuous in Θ, can be optimized by gradient methods. Acc is not continuous in Θ and is unsuited to direct optimization (indeed, finding an optimum is NP-complete).

When optimizing an arbitrary function of Θ, we have to make sure that our probabilities remain well-formed. If we want to have a well-formed joint NB interpretation, we must have non-negative parameters and the inequalities $\forall s \sum_o \theta_{o|s} \leq 1$ and $\sum_s \theta_s \leq 1$. If we want to be guaranteed a non-deficient joint interpretation, we can require equality. However, if we relax the equality then we have a larger feasible space which may give better values of our objective.

We performed the following WSD experiments with Naive-Bayes models. We took as data the collection of SENSEVAL-2 English lexical sample WSD corpora.[2] We set the NB model parameters in several ways. We optimized JL (using the RFEs).[3] We also optimized SCL and (the log of) CL, using a conjugate gradient (CG) method (Press et al., 1988).[4] For CL and SCL, we optimized each objective both over the space of all distributions and over the subspace of non-deficient models (giving CL* and SCL*). Acc was not directly optimized.

Unconstrained CL corresponds exactly to a conditional maximum entropy model (Berger et al., 1996; Lafferty et al., 2001). This particular case, where there are multiple explanatory variables and a single categorical response variable, is also precisely the well-studied statistical model of (multinomial) *logistic regression* (Agresti, 1990). Its optimization problem is concave (over log parameters) and therefore has a unique global maximum. For CL*, SCL, and SCL*, we are only guaranteed local optima, but in practice we detected no maxima which were not global over the feasible region.

TRAINING SET					
Optimization	Acc	MacroAcc	$\log JL$	$\log CL$	SCL
NB-JL	86.8	86.2	**-22969684.7**	-243184.1	4505.9
NB-CL*	**98.5**	**96.2**	-23366291.2	-973.0	5101.2
NB-CL	**98.5**	**96.2**	-23431010.0	**-854.1**	**5115.1**
NB-SCL*	94.2	93.7	-23054768.6	-226187.8	4884.4
NB-SCL	97.3	95.5	-23146735.3	-220145.0	5055.8

TEST SET					
Optimization	Acc	MacroAcc	$\log JL$	$\log CL$	SCL
NB-JL	73.6	55.0	**-1816757.1**	-55251.5	3695.4
NB-CL*	72.3	53.4	-1954977.1	**-19854.1**	3566.3
NB-CL	76.2	56.5	-1964498.5	-20498.7	3798.8
NB-SCL*	74.8	57.2	-1841305.0	-43027.8	3754.1
NB-SCL	**77.5**	**59.7**	-1872533.0	-33249.7	**3890.8**

Figure 2: Scores for the NB model trained according to various objectives. Scores are usually higher on both training and test sets for the objective maximized, and discriminative criteria lead to better test-set accuracy. The best scores are in bold.

Figure 2 shows, for each objective maximized, the values of all objectives on both the training and test set. Optimizing for a given objective generally gave the best score for that objective for both the training set and the test set. The exception is NB-SCL and NB-SCL* which have lower SCL score than NB-CL and NB-CL*. This is due to the penalty used for smoothing the summed models (see fn. 3).

Accuracy is higher when optimizing the discriminative objectives, CL and SCL, than when optimizing JL (including for macro-averaging, where each word's contribution to average accuracy is made equal). That these estimates beat NB-JL on accuracy is unsurprising, since Acc is a discretization of conditional predictions, not joint ones. This supports the claim that maximizing conditional likelihood, or other discriminative objectives, improves test set accuracy for realistic NLP tasks. NB-SCL, though harder to maximize in general, gives better test-set accuracy than NB-CL.[5] NB-CL* is somewhere between JL and CL for all objectives on the training data. Its behavior shows that the change from a standard NB approach (NB-JL) to a maximum entropy classifier (NB-CL) can be broken into two aspects: a change in objective and an abandonment of a non-deficiency constraint.[6] Note that the JL score for NB-CL*, is not very much lower than for NB-JL, despite a large change in CL.

It would be too strong to state that maximizing CL

[2] http://www.sle.sharp.co.uk/senseval2/

[3] Smoothing is an important factor for this task. So that the various estimates would be smoothed as similarly as possible, we smoothed implicitly, by adding smoothing data. We added one instance of each class occurring with the bag containing each vocabulary word once. This gave the same result as add-one smoothing on the RFEs for NB-JL, and ensured that NB-CL would not assign zero conditional probability to any unseen event. The smoothing data did not, however, result in smoothed estimates for SCL; any conditional probability will sum to one over the smoothing instances. For this objective, we added a penalty term proportional to $\sum \theta^2$, which ensured that no conditional sense probabilities reached 0 or 1.

[4] All optimization was done using conjugate gradient ascent over log parameters $\lambda_i = \log \theta_i$, rather than the given parameters due to sensitivity near zero and improved quality of quadratic approximations during optimization. Linear constraints over θ are not linear in log space, and were enforced using a quadratic Lagrange penalty method (Bertsekas, 1995).

[5] This difference seems to be partially due to the different smoothing methods used: Chen and Rosenfeld (1999) show that quadratic penalties are very effective in practice, while the smoothing-data method is quite crude.

[6] If one is only interested in the model's conditional predictions, there is no reason to disprefer deficient joint models.

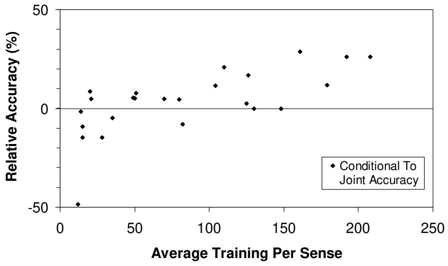
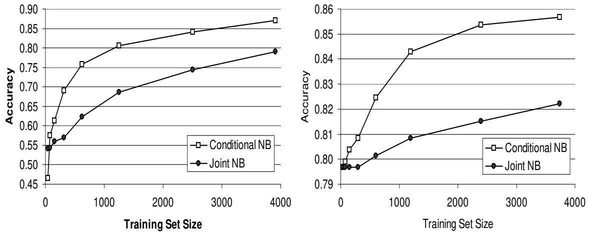


Figure 3: Conditional NB has higher accuracy than joint NB for WSD on most SENSEVAL-2 word sets. The relative improvement gained by switching to conditional estimation is positively correlated to training set size.



(a) (b)

Figure 4: Conditional NB is better than Joint NB for WSD given all but possibly the smallest training sets, and the advantage increases with training set size. (a) "line" (b) "hard"

(in particular) and discriminative objectives (in general) is *always* better than maximizing JL for improving test-set accuracy. Even on the present task, CL strictly beat JL in accuracy for only 15 of 24 words. Figure 3 shows a plot of the relative accuracy for CL: $(\mathrm{Acc_{CL}} - \mathrm{Acc_{JL}})/\mathrm{Acc_{JL}}$. The x-axis is the average number of training instances per sense, weighted by the frequency of that sense in the test data. There is a clear trend that larger training sets saw a larger benefit from using NB-CL. The scatter in this trend is partially due to the wide range in data set conditions. The data sets exhibit an unusual amount of drift between training and test distributions. For example, the test data for *amaze* consists entirely of 70 instances of the *less* frequent of its two training senses, and represents the highest point on this graph, with NB-CL having a relative accuracy increase of 28%. This drift between the training and test corpora generally favors conditional estimates. On the other hand, many of these data sets are very small, individually, and 6 of the 7 sets where NB-JL wins are among the 8 smallest, 4 of them in fact being the 4 smallest. Ng and Jordan (2002) show that, between NB-JL and NB-CL, the discriminative NB-CL should, in principle, have a lower asymptotic error, but the generative NB-JL should perform better in low-data situations. They argue that unless one has a relatively large data set, one is in fact likely to be better off with the generative estimate. Their claim seems too strong here; even smaller data sets often show benefit to accuracy from CL estimation, although all would qualify as small on their scale.

Since the number of senses and skew towards common senses is so varied between SENSEVAL-2 words, we turned to larger data sets to test the effective "break-even" size for WSD data, using the *hard* and *line* data from Leacock et al. (1998). Figure 4 shows the accuracy of NB-CL and NB-JL as the amount of training data increases. Conditional beats joint for all but the smallest training sizes, and the improvement is greater with larger training sets. Only for the *line* data does the conditional model *ever* drop below the joint model.

For this task, then, NB-CL is performing better than expected. This appears to be due to two ways in which CL estimation is suited to linguistic data. First, the Ng and Jordan results do not involve smoothed data. Their data sets do not require it like linguistic data does, and smoothing largely prevents the low-data overfitting that can plague conditional models.

There is another, more interesting reason why conditional estimation for this model might work better for an NLP task like WSD than for a general machine learning task. One signature difficulty in NLP is that the data contains a great many rare observations. In the case of WSD, the issue is in telling the kinds of rare events apart. Consider a word w which occurs only once, with a sense s. In the joint model, smoothing ensures that w does not signal s too strongly. However, every w which occurs only once with s will receive the same $P(w|s)$. Ideally, we would want to be able to tell the accidental singletons from true indicator words. The conditional model implicitly does this to a certain extent. If w occurs with s in an example where other good indicator words are present, then those other words' large weights will explain the occurrence of s, and without w having to have a large weight, its expected count with s in that instance will approach 1. On the other hand, if no trigger words occur in that instance, there will be no other explanation for s other than the presence of w and the other non-indicative words. Therefore, w's weight, and the other words', will grow until s is predicted sufficiently strongly.

As a concrete illustration, we isolated two senses of "line" into a two-sense data set. Sense 1 was "a queue" and sense 2 was "a phone line." In this corpus, the words *transatlantic* and *flowers* both occur only once, and only with the "phone" sense (plus once with each in the smoothing data). However, *transatlantic* occurs in the instance *thanks, anyway, the transatlantic line_2 died.* , while *flowers* occurs in the longer instance … *phones with more than one line_2, plush robes, exotic flowers, and complimentary wine.* In the first instance, the only non-singleton content word is *died* which occurs once with sense 1 and twice with sense 2. However, in the other case, *phone* occurs 191 times with sense 2 and only 4 times with sense 1. Additionally, there are more words in the second instance with which *flowers* can share the burden of increasing its expectation. Experimentally,

$$\frac{P_{\mathrm{JL}}(\textit{flowers}|2)}{P_{\mathrm{JL}}(\textit{flowers}|1)} = \frac{P_{\mathrm{JL}}(\textit{transatlantic}|2)}{P_{\mathrm{JL}}(\textit{transatlantic}|1)} = 2$$

while with conditional estimation,

$$\frac{P_{\mathrm{CL}}(\textit{flowers}|2)}{P_{\mathrm{CL}}(\textit{flowers}|1)} = 2.05$$

$$\frac{P_{\mathrm{CL}}(\textit{transatlantic}|2)}{P_{\mathrm{CL}}(\textit{transatlantic}|1)} = 3.74$$

With joint estimation, both words signal sense 2 with equal strength. With conditional estimation, the presense of words like *phone* cause *flowers* to indicate sense 2 less strongly that *transatlantic*. Given that the conditional estimation is implicitly differentially weighting rare events in a plausibly way, it is perhaps unsurprising that a task like WSD would see the benefits on smaller corpus sizes than would be expected on standard machine-learning data sets.[7]

These trends are reliable, but sometimes small. In practice, one must decide if, for example, a 5% error reduction is worth the added work: CG optimization, especially with constraints, is considerably harder to implement than simple RFE estimates for JL. It is also considerably slower: the total training time for the entire SENSEVAL-2 corpus was less than 3 seconds for NB-JL, but two hours for NB-CL.

3 Model Structure: HMMs and CMMs

We now consider sequence data, with POS tagging as a concrete NLP example. In the previous section, we had a single model, but several ways of estimating parameters. In this section, we have two different model structures.

First is the classic hidden Markov model (HMM), shown in figure 6a. For an instance $(\mathbf{s}, \mathbf{o})$, where

[7] Interestingly, the common approach of discarding low-count events (for both training speed and overfitting reasons) when estimating the conditional models used in maxent taggers robs the system of the opportunity to exploit this effect of conditional estimation.

	Model		
Objective	HMM	MEMM	MEMM†
JL	91.23	89.22	90.44
CL*	91.41	89.22	90.44
CL	91.44	89.22	90.44

Figure 5: Tagging accuracy: For a fixed model, conditional estimation is slightly advantageous. For a fixed objective, the MEMM is inferior, though it can be improved by *unobserving* unambiguous words.

$\mathbf{o} = \langle o_i \rangle$ is a word sequence and $\mathbf{s} = \langle s_i \rangle$ is a tag sequence, we write the following (joint) model:

$$P(\mathbf{s}, \mathbf{o}) = P(\mathbf{s})P(\mathbf{o}|\mathbf{s}) = \prod_i P(s_i|s_{i-1})P(o_i|s_i)$$

where we use a start state s_0 to simplify notation.

The parameters of this model are the transition and emission probabilities. Again, we can set these parameters to maximize JL, as is typical, or we can set them to maximize other objectives, without changing the model *structure*. If we maximize CL, we get (possibly deficient) HMMs which are instances of the conditional random fields of Lafferty et al. (2001).[8]

Figure 5 shows the tagging accuracy of an HMM trained to maximize each objective. JL is the standard HMM. CL duplicates the simple CRFs in (Lafferty et al., 2001). CL* is again an intermediate, where we optimized conditional likelihood but required the HMM to be non-deficient. This separates out the benefit of the conditional objective from the benefit from the possibility of deficiency (which relates to label bias, see below). In accordance with our observations in the last section, and consistent with the results of (Lafferty et al., 2001), the CL accuracy is slightly higher than JL for this fixed model.

Another model often used for sequence data is the upward Conditional Markov Model (CMM), shown as a graphical model in figure 6b. This is the model used in maximum entropy tagging. The graphical model shown gives a *joint* distribution over $(\mathbf{s}, \mathbf{o})$, just like an HMM. It is a conditionally *structured* model, in the sense that that distribution can be written as $P(\mathbf{s}, \mathbf{o}) = P(\mathbf{s}|\mathbf{o})P(\mathbf{o})$. Since tagging only uses $P(\mathbf{s}|\mathbf{o})$, we can discard what the model says about $P(\mathbf{o})$. The model as drawn assumes that each observation is independent, but we could add any arrows we please among the o_i without changing the conditional predictions. Therefore, it is common to think about this model as if the joint interpretation were absent, and not to model the observations at all. For models which are conditional in the sense of

[8] The general class of CRFs is more expressive and reduces to deficient HMMs only when they have just these features.

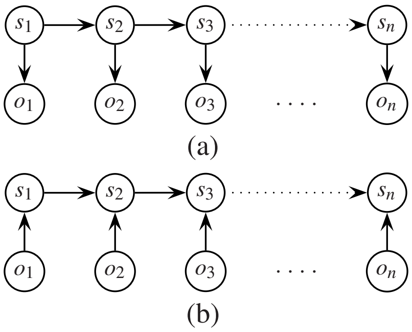


Figure 6: Graphical models: (a) the downward HMM, and (b) the upward conditional Markov model (CMM).

the factorization above, the JL and CL estimates for $P(\mathbf{s}|\mathbf{o})$ will always be the same. It is therefore tempting to believe that since one can find closed-form CL estimates (the RFEs) for these models, one can gain the benefit of conditional estimation. We will show that this is not true, at least not here.

Adopting the CMM has effects in and of itself, regardless of whether a maximum entropy approach is used to populate the $P(s|s_{-1}, o)$ estimates. The ML estimate for this model is the RFE for $P(s|s_{-1}, o)$. For tagging, sparsity makes this impossible to reliably estimate directly, but even if we could do so, we would have a graphical model with several defects. Every graphical model embodies conditional independence assumptions. The NB model assumes that observations are independent given the class. The HMM assumes the Markov property that future observations are independent from past ones given the intermediate state. Both assumptions are obviously false in the data, but the models do well enough for the tasks we ask of them. However, the assumptions in this upward model are worse, both qualitatively and quantitatively. It is a conditional model, in that the model can be factored as $P(\mathbf{o})P(\mathbf{s}|\mathbf{o})$. As a result, it makes no useful statement about the distribution of the data, making it useless, for example, for generation or language modeling. But more subtly note that states are independent of future observations. As a result, future cues are unable to influence past decisions in certain cases. For example, imagine tagging an entire sentence where the first word is an unknown word. With this model structure, if we ask about the possible tags for the first word, we will get back the marginal distribution over (sentence-initial) unknown words' tags, regardless of the following words.

We constructed two taggers. One was an HMM, as in figure 6a. It was trained for JL, CL*, and CL. The second was a CMM, as in figure 6b. We used a maximum entropy model over the (word, tag) and (previous-tag, tag) features to approximate the $P(s|s_{-1}, o)$ conditional probabilities. This CMM is referred to as an MEMM. A 9-1 split of the Penn treebank was used as the data corpus. To smooth these models as equally as possible and to give as unified a treatment of unseen words as possible, we mapped all words which occurred only once in training to an unknown token. New words in the test data were also mapped to this token.[9]

Using these taggers, we examined what kinds of errors actually occurred. One kind of error tendency in CMMs which has been hypothesized in the literature is called *label bias* (Bottou, 1991; Lafferty et al., 2001). Label bias is a type of explaining-away phenomenon (Pearl, 1988) which can be attributed to the local conditional modeling of each state. The idea is that states whose following-state distributions have low entropy will be preferred. Whatever mass arrives at a state must be pushed to successor states; it cannot be dumped on alternate observations as in an HMM. In theory, this means that the model can get into a dysfunctional behavior where a trajectory has no relation to the observations but will still stumble onward with high conditional probability. The sense in which this is an explaining-away phenomenon is that the previous state explains the current state so well that the observation at the current state is effectively ignored. What we found in the case of POS tagging was the opposite. The state-state distributions are on average nowhere near as sharply distributed as the state-observation distributions. This gives rise to the reverse explaining-away effect. The observations explain the states above them so well that the previous states are effectively ignored. We call this *observation bias*.

As an example, consider what happens when a word has only a single tag. The conditional distribution for the tag above that word will always assign conditional probability one to that single tag, regardless of the previous tag. Figure 7 shows the sentence *All the indexes dove* ., in which *All* should be tagged as a predeterminer (PDT).[10] Most occurrences of *All*, however, are as a determiner (DT, 106/135 vs 26/135), and it is much more common for a sentence to begin with a determiner than a predeterminer. The

[9]Doing so lowered our accuracy by approximately 2% for all models, but gave better-controlled experiments.

[10]The treebank predeterminer tag is meant for when words like *All* are followed by a determiner, as in this case.

						HMM	MEMM	MEMM†
Correct States	PDT	DT	NNS	VBD	.	-0.0	-1.3	-0.0
Incorrect States	DT	DT	NNS	VBD	.	-5.4	-0.3	-5.7
Observations	All	the	indexes	dove	.			

Figure 7: The MEMM exhibits observation bias: knowing that *the* is a DT makes the quality of the DT-DT transition irrelevant, and *All* receives its most common tag (DT).

other words occur with only one tag in the treebank.[11] The HMM tags this sentence correctly, because two determiners in a row is rarer than *All* being a predeterminer (and a predeterminer beginning a sentence). However, the MEMM shows exactly the effect described above, choosing the most common tag (DT) for *All*, since the choice of tag for *All* does not effect the conditional tagging distribution for *the*. The MEMM parameters do assign a lower weight to the DT DT feature than to the PDT DT feature, but the *the* ensures a DT tag, regardless.

Exploiting the joint interpretation of the CMM, what we can do is to *unobserve* word nodes, leaving the graphical model as it is, but changing the observation status of a given node to "not observed". For example, we can retain our knowledge that the state above *the* is DT, but "forget" that we know that the word at that position is *the*. If we do inference in this example with *the* unobserved, taking a weighted sum over all values of that node, then the conditional distribution over tag sequences changes as shown under MEMM†: the correct tagging has once again become most probable. Unobserving the word itself is not *a priori* a good idea. It could easily put too much pressure on the last state to explain the fixed state. This effect is even visible in this small example: the likelihood of the more typical PDT-DT tag sequence is even higher for MEMM† than the HMM.

These issues are quite important for NLP, since state-of-the-art statistical taggers are all based on one of these two models. In order to check which, if either, of label or observation bias is actually contributing to tagging error, we performed the following experiments with our simple HMM and MEMM taggers. First, we measured, on the training data, the entropy of the next-state distribution $P(s|s_{-1})$ for each state s. For both the HMM and MEMM, we then measured the relative overproposal rate for each state: the number of errors where that state was incorrectly predicted in the test set, divided by the overall frequency of that state in the correct answers. The label bias hypothesis makes a concrete prediction: lower entropy

[11]For the sake of clarity, this example has been slightly doctored by the removal of several non-DT occurrences of *the* in the treebank – all incorrect.

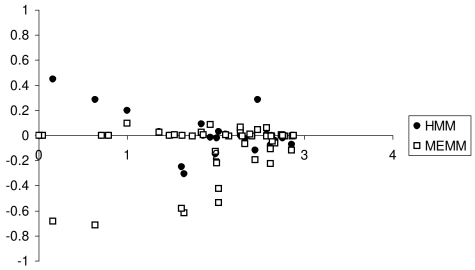


Figure 8: State transition entropy (x-axis) does not appear to be positively correlated with the relative over-proposal frequency (y-axis) of the tags for the MEMM model, though it is slightly so with the HMM model.

states should have higher relative overproposal values, especially for the MEMM. Figure 8 shows that the trends, if any, are not clear. There does appear to be a slight tendency to have higher error on the low-entropy tags for the HMM, but if there is any superficial trend for the MEMM, it is the reverse.

On the other hand, if systematically unobserving unambiguous observations in the MEMM led to an increase in accuracy, then we would have evidence of observation bias. Figure 5 shows that this is exactly the case. The error rate of the MEMM drops when we unobserve these single-tag words (from 10.8% to 9.5%), and the error rate in positions before such words drops even more sharply (17.1% to 15.0%). The drop in overall error in fact cuts the gap between the HMM and the MEMM by about half.

The claim here is not that label bias is impossible for MEMMs, nor that state-of-the-art maxent taggers would necessarily benefit from the unobserving of fixed-tag words – if there are already (tag, next-word) features in the model, this effect should be far weaker. The claim is that the independence assumptions embodied by the conditionally structured model were the primary root of the lower accuracy for this model. Label bias and observation bias are both explaining-away phenomena, and are both consequences of these assumptions. Explaining-away effects will be found quite generally in conditionally-structured models, and should be carefully considered before such models are adopted. The effect can be good or bad: In the case of the NB-CL model, there was also an explaining-away effects among the words. This is exactly the cause for *flowers* being a weaker indicator than *transatlantic* in our conditional estimation example. In that case, we *wanted* certain word occurrences to be explained away by the presence of more explanatory words. However, when some of the competing conditioned features are previous local decisions, ignoring them can be harmful.

4 Related Results

Johnson (2001) describes two parsing experiments. First, he examines a PCFG over the ATIS treebank, trained both using RFEs to maximize JL, and using a CG method to maximize what we have been calling CL*. He does not give results for the unconstrained CL, but even in the constrained case, the effects from section 2 occur. CL and parsing accuracy are both higher using the CL* estimates. He also describes a conditional shift-reduce parsing model, but notes that it underperforms the simpler joint formulation. We take these two results not as contradictory, but as confirmation that conditional estimation, though often slow, generally improves accuracy, while conditional model structures must be used with caution. The conditional shift-reduce parsing model he describes can be expected to exhibit the same type of competing-variable explaining-away issues that occur in MEMM tagging. As an extreme example, if all words have been shifted, the rest of the parser actions will be reductions with probability one.

Goodman (1996) describes algorithms for parse selection where the criterion being maximized in parse selection is the bracket-based accuracy measure that parses are scored by. He shows a test-set accuracy benefit from optimizing accuracy directly.

Finally, model structure and parameter estimation are not the entirety of factors which determine the behavior of a model. Model features are crucial, and the ability to incorporate richer features in a relatively sensible way also leads to improved models. This is the main basis of the real world benefit which has been derived from maxent models.

5 Conclusions

We have argued that optimizing an objective that is as close to the task "accuracy" as possible is advantageous in NLP domains, even in data-poor cases where machine-learning results suggest discriminative approaches may not be reliable. We have also argued that the model structure is a far more important issue. For simple POS tagging, the observation bias effect of the model's independence assumptions is more evident than label bias as a source of error, but both are examples of explaining-away effects which can arise in conditionally structured models. Our results, combined with others in the literature, suggest that conditional model structure is, in and of itself, undesirable, unless that structure enables methods of incorporating better features, explaining why maximum-entropy taggers and parsers have had such success despite the inferior performance of their basic skeletal models.

References

Alan Agresti. 1990. *Categorical Data Analysis*. John Wiley & Sons, New York.

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

D. P. Bertsekas. 1995. *Nonlinear Programming*. Athena Scientific, Belmont, MA.

Léon Bottou. 1991. *Une approche theorique de l'apprentissage connexioniste; applications a la reconnaissance de la parole*. Ph.D. thesis, Université de Paris XI.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *ANLP 6*, pages 224–231.

S. Chen and R. Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical Report CMU CS-99-108, Carnegie Mellon University.

Jerome H. Friedman. 1997. On bias, variance, 0/1–loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

Joshua Goodman. 1996. Parsing algorithms and metrics. In *ACL 34*, pages 177–183.

Mark Johnson. 2001. Joint and conditional estimation of tagging and parsing models. In *ACL 39*, pages 314–321.

A. Kilgarriff. 1998. Senseval: An exercise in evaluating word sense disambiguation programs. In LREC, pages 581–588.

John Lafferty, Fernando Pereira, and Andrew McCallum. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and Wordnet relations for sense identification. *Computational Linguistics*, 24:147–165.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *Working Notes of the 1998 AAAI/ICML Workshop on Learning for Text Categorization*.

Andrew Y. Ng and Michael Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS 14*.

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

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1988. *Numerical Recipes in C*. Cambridge University Press, Cambridge.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Scott M. Thede and Mary P. Harper. 1999. Second-order hidden Markov model for part-of-speech tagging. In *ACL 37*, pages 175–182.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *EMNLP/VLC 2000*, pages 63–70.