Structured Local Training and Biased Potential Functions for Conditional Random Fields with Application to Coreference Resolution

Yejin Choi and Claire Cardie
Department of Computer Science
Cornell University
Ithaca, NY 14853
{ychoi,cardie}@cs.cornell.edu

Abstract

Conditional Random Fields (CRFs) have shown great success for problems involving structured output variables. However, for many real-world NLP applications, exact maximum-likelihood training is intractable because computing the global normalization factor even approximately can be extremely hard. In addition, optimizing likelihood often does not correlate with maximizing task-specific evaluation measures. In this paper, we present a novel training procedure, *structured local training*, that maximizes likelihood while exploiting the benefits of global inference during training: hidden variables are used to capture interactions between local inference and global inference. Furthermore, we introduce *biased potential functions* that empirically drive CRFs towards performance improvements w.r.t. the preferred evaluation measure for the learning task. We report promising experimental results on two coreference data sets using two task-specific evaluation measures.

1 Introduction

Undirected graphical models such as Conditional Random Fields (CRFs) (Lafferty et al., 2001) have shown great success for problems involving structured output variables (e.g. Wellner et al. (2004), Finkel et al. (2005)). For many real-world NLP applications, however, the required graph structure can be very complex, and computing the global normalization factor even approximately can be extremely hard. Previous approaches for training CRFs have either (1) opted for a training method that no longer maximizes the likelihood, (e.g. McCallum and Wellner (2004), Roth and Yih (2005)) [1], or (2) opted for a simplified graph structure to avoid intractable global normalization (e.g. Roth and Yih (2005), Wellner et al. (2004)).

Solutions of the first type replace the computation of the global normalization factor $\sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$ with $\text{argmax}_{\mathbf{y}}\ p(\mathbf{y}|\mathbf{x})$ during training, since finding an argmax of a probability distribution is often an easier problem than finding the entire probability distribution. Training via the voted perceptron algorithm (Collins, 2002) or using a max-margin criterion also correspond to the first option (e.g. McCallum and Wellner (2004), Finley and Joachims (2005)). But without the global normalization, the maximum-likelihood criterion motivated by the maximum entropy principle (Berger et al., 1996) is no longer a feasible option as an optimization criterion.

The second solution simplifies the graph structure for training, and applies complex global inference only for testing. In spite of the discrepancy between the training model and the testing model, it has been empirically shown that (1) performing global inference only during testing can improve performance (e.g. Finkel et al. (2005), Roth and Yih (2005)), and (2) full-blown global training can often perform worse due to insufficient training data (e.g. Punyakanok et al. (2005)). Importantly, however, attempts to reduce the discrepancy between the training and test models — by judiciously adding the effect of global inference to the training — have produced substantial performance improvements over locally trained models (e.g. Cohen and Carvalho (2005), Sutton and McCallum (2005a)).

In this paper, we present *structured local training*, a novel training procedure for maximum-likelihood

[1] Both McCallum and Wellner (2004) and Roth and Yih (2005) used the voted perceptron algorithm (Collins, 2002) to train intractable CRFs.

Proceedings of NAACL HLT 2007, pages 65–72,
Rochester, NY, April 2007.

training of undirected graphical models, such as CRFs. The procedure maximizes likelihood while exploiting the benefits of global inference during training by capturing the interactions between local inference and global inference via hidden variables.

Furthermore, we introduce *biased potential functions* that redefine the likelihood for CRFs so that the performance of CRFs trained under the maximum likelihood criterion correlates better empirically with the preferred evaluation measures such as F-score and MUC-score.

We focus on the problem of coreference resolution; however, our approaches are general and can be extended to other NLP applications with structured output. Our approaches also extend to non-conditional graphical models such as Markov Random Fields. In experiments on two coreference data sets, structured local training reduces the error rate significantly (3.5%) for one coreference data set and minimally ($\leq$ 1%) for the other. Experiments using biased potential functions increase recall uniformly and significantly for both data sets and both task-specific evaluation measures. Results for the combination of the two techniques are promising, but mixed: pairwise F1 increases by 0.8-5.5% for both data sets; MUC F1 increases by 3.5% for one data set, but slightly hurts performance for the second data set.

In §2, we describe *structured local training*, and follow with experimental results in §3. In §4, we describe *biased potential functions* and follow with experimental results in §5. We discuss related work in §6.

2 Structured Local Training

2.1 Definitions

For clarity, we define the following terms that we will use throughout the paper.

- **local inference:** [2] Inference factored into smaller independent pieces, without considering the structure of the output space.
- **global inference:** Inference applied on the entire set of output variables, considering the structure of the output space.
- **local training:** Training that does not invoke global inference at each iteration.
- **global training:** Training that does invoke global inference at each iteration.

2.2 A Motivating Example for Coreference Resolution

In this section, we present an example of the coreference resolution problem to motivate our approach. It has been shown that global inference-based training for coreference resolution outperforms training with local inference only (e.g. Finley and Joachims (2005), McCallum and Wellner (2004)). In particular, the output of coreference resolution must obey equivalence relations, and exploiting such structural constraints on the output space during training can improve performance. Consider the coreference resolution task for the following text.

> It was after the passage of this act, that Mary$^{(1)}$'s attitude towards Elizabeth$^{(1)}$ became overtly hostile. The deliberations surrounding the act seem to have revived all Mary's memories of the humiliations she had suffered at the hands of Anne Boleyn. At the same time, Elizabeth$^{(2)}$'s continuing prevarications over religion confirmed that **she** was indeed her mother's daughter.

In the above text, the "she" in the last sentence is coreferent with both mentions of "$Elizabeth$". However, when we consider "she" and "$Elizabeth^{(1)}$" in isolation from the remaining coreference chain, it can be difficult for a machine learning method to determine whether the pair is coreferent or not. Indeed, such a pair may not look very different from the pair "she" and "$Mary^{(1)}$" in terms of feature vectors. It is much easier, however, to determine that "she" and "$Elizabeth^{(2)}$" are coreferent, or that "$Elizabeth^{(1)}$" and "$Elizabeth^{(2)}$" are coreferent. Only by taking the transitive closure of these pairwise coreference relations does it become clear that "she" and "$Elizabeth^{(1)}$" are coreferent. In other words, global training might handle potentially confusing coreference cases better because it allows parameter learning (for each pairwise coreference decision) to be informed by global inference.

We argue that, with appropriate modification to the learning instances, local training is adequate for the coreference resolution task. Specifically, we propose that confusing pairs in the training data — such

[2] In this paper, inference refers to the operation of finding the *argmax* in particular.

as "she" and "$Elizabeth^{(1)}$" — be learned as *not-coreferent*, so long as the global inference step can fix this error by exploiting the structure of the output space, i.e. by exploiting the equivalence relations. This is the key idea of *structured local training*, which we elaborate formally in the following section.

2.3 A Hidden-Variable Model

In this section, we present a general description of *structured local training*. Let **y** be a vector of output variables for structured output, and let **x** be a vector of input variables. In order to capture the interactions between global inference and local inference, we introduce hidden variables **h**, $|\mathbf{h}| = |\mathbf{y}|$, so that the global inference for $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ can be factored into two components using the product rule, as follows:

$$\begin{aligned} p(\mathbf{y}, \mathbf{h}|\mathbf{x}) &= p(\mathbf{y}|\mathbf{h}, \mathbf{x})\, p(\mathbf{h}|\mathbf{x}) \\ &= p(\mathbf{y}|\mathbf{h})\, p(\mathbf{h}|\mathbf{x}) \end{aligned}$$

The second component $p(\mathbf{h}|\mathbf{x})$ on the right hand side corresponds to the local model, for which the inference factorizes into smaller independent pieces, e.g. $\text{argmax}_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) = \{\text{argmax}_{h_i} \phi(h_i, \mathbf{x})\}$. And the first component $p(\mathbf{y}|\mathbf{h}, \mathbf{x})$ on the right hand side corresponds to the global model, whose inference may not factorize nicely. Further, we assume that **y** is independent of **x** given **h**, so that $p(\mathbf{y}|\mathbf{h}, \mathbf{x}) = p(\mathbf{y}|\mathbf{h})$. That is to say, **h** captures sufficient information from **x**, so that given **h**, global inference of **y** only depends on **h**. The quantity of $p(\mathbf{y}|\mathbf{x})$ then is given by marginalizing out **h** as follows:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y}, \mathbf{h}|\mathbf{x})$$

Intuitively, the hidden variables **h** represent the local decisions that can lead to a good **y** after global inference is applied. In the case of coreference resolution, one natural factorization would be that global inference is a clustering algorithm, and local inference is a classification decision on each pair of noun phrases (or mentions).[3] In this paper, we assume that we only parameterize the local model $p(\mathbf{h}|\mathbf{x})$, although it would be possible to extend the parameterization to the global model as well, depending on the particular application under consideration. The similarity between a pair of mentions is parameterized via log-linear models. However, once we have the similarity scores extracted via local inference, the clustering algorithm does not require further parameterization.

For training, we apply the standard Expectation-Maximization (EM) algorithm (Dempster et al., 1977) as follows:

- E Step: Compute a distribution

$$\widetilde{P}^{(t)} = P(\mathbf{h}|\mathbf{y}, \mathbf{x}, \theta^{(t-1)})$$

- M Step: Set $\theta^{(t)}$ to θ that maximizes

$$E_{\widetilde{P}^{(t)}} \left[\log P(\mathbf{y}, \mathbf{h}|\mathbf{x}, \theta)\right]$$

By repeatedly applying the above two steps for $t = 1, 2, ...$, the value of θ converges to the local maxima of the conditional log likelihood $L(\theta) = \log P(\mathbf{y}|\mathbf{x}, \theta)$.

2.4 Application to Coreference Resolution

For $y_i \in \mathbf{y}$ (and $h_i \in \mathbf{h}$) in the coreference resolution task, $y_i = 1$ (and $h_i = 1$) corresponds to ith pair of mentions being coreferent, and $y_i = 0$ (and $h_i = 0$) corresponds to ith pair being not coreferent.

[Local Model $P(\mathbf{h}|\mathbf{x})$] For the local model, we define cliques as individual nodes,[4] and parameterize each clique potential as

$$\phi(h_i, \mathbf{x}) = \phi(h_i, x_i) = \exp \sum_k \lambda_k f_k(h_i, x_i)$$

Let $\Phi(\mathbf{h}|\mathbf{x}) \equiv \prod_i \phi(h_i, x_i)$. Then,

$$P(\mathbf{h}|\mathbf{x}) = \frac{\Phi(\mathbf{h}, \mathbf{x})}{\sum_{\mathbf{h}} \Phi(\mathbf{h}, \mathbf{x})}$$

Notice that in this model, finding $\text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})$ corresponds to simply finding $\text{argmax}_{h_i} \phi(h_i, x_i)$ independently for each $h_i \in \mathbf{h}$.

[3] Formally, we define each $y_i \in \mathbf{y}$ to be the coreference decision for the ith pair of mentions, and $x_i \in \mathbf{x}$ be the input regarding the ith pair of mentions. Then h_i corresponds to the local coreference decision that can lead to a good coreference decision y_i after the clustering algorithm has been applied.

[4] Each node in the graphical representation of CRFs corresponds to the coreferent decision for each pair of mentions. This corresponds to the "Model 3" of McCallum and Wellner (2004).

ALGORITHM-1

INPUT: $\mathbf{x}$, true labeling $\mathbf{y}^*$, current local model $P(\mathbf{h}|\mathbf{x})$
GOAL: Find the highest confidence labeling $\mathbf{y}'$
such that $\mathbf{y}^*$ = single-link-clustering($\mathbf{y}'$)

$\mathbf{h}^* \leftarrow \text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})$
$\mathbf{h}' \leftarrow$ single-link-clustering($\mathbf{h}^*$)
construct a graph $G = (V, E)$, where
  $E = \{h'_i : h'_i \in \mathbf{h}' \text{ s.t. } y^*_i = 1\}$
  $V = \{v : v \text{ is a NP referred by a } h'_i \in E\}$
  with edge cost $cost_{h'_i} = \phi(h'_i, x_i)$ if $h'_i \neq y^*_i$
  with edge cost $cost_{h'_i} = 0$ if $h'_i = y^*_i$
find a minimum spanning tree(or forest) M of G
for each $h'_i \in \mathbf{h}'$
  if $h'_i = y^*_i$
    $y'_i \leftarrow h^*_i$
  else if $h'_i \in M$
    $y'_i \leftarrow 1$
  else
    $y'_i \leftarrow 0$
end for
return $\mathbf{y}'$

Figure 1: Algorithm to find the highest confidence labeling $\mathbf{y}'$ that can be clustered to the true labeling $\mathbf{y}^*$

[Global Model $P(\mathbf{y}|\mathbf{h})$] For the global model, we assume a deterministic clustering algorithm is given. In particular, we focus on single-link clustering, as it has been shown to be effective for coreference resolution (e.g. Ng and Cardie (2002)). With single-link clustering, $P(\mathbf{y}|\mathbf{h}) = 1$ if $\mathbf{h}$ can be clustered to $\mathbf{y}$, and $P(\mathbf{y}|\mathbf{h}) = 0$ if $\mathbf{h}$ cannot be clustered to $\mathbf{y}$.[5]

[Computation of the E-step] The E-step requires computation of the distribution of $P(\mathbf{h}|\mathbf{y}, \mathbf{x}, \theta^{(t-1)})$, which we will simply denote as $P(\mathbf{h}|\mathbf{y}, \mathbf{x})$, since all our distributions are implicitly conditioned on the model parameters θ.

$$P(\mathbf{h}|\mathbf{y}, \mathbf{x}) = \frac{P(\mathbf{h}, \mathbf{y}|\mathbf{x})}{P(\mathbf{y}|\mathbf{x})} \propto P(\mathbf{y}|\mathbf{h})\ P(\mathbf{h}|\mathbf{x})$$

Notice that when computing $P(\mathbf{h}|\mathbf{y}, \mathbf{x})$, the denominator $P(\mathbf{y}|\mathbf{x})$ stays as a constant for different values of $\mathbf{h}$. The E-step requires enumeration of all possible values of $\mathbf{h}$, but it is intractable with our formulation, because inference for the global model $P(\mathbf{y}|\mathbf{h})$ does not factor out nicely. Therefore, we must resort to an approximation method. Neal and Hinton (1998) analyze and motivate various approximate EM training methods. One popular choice in practice is called "Viterbi training", a variant of the EM algorithm, which has been shown effective in many NLP applications. Viterbi training approximates the distribution by assigning all probability mass to a single best assignment. The algorithm for this is shown in Figure 1.

[5] Single-link clustering simply takes the transitive closure, and does not consider the distance metric. In a pilot study, we also tried a variant of a stochastic clustering algorithm that takes into account the distance metric (set as the probabilities from the local model) for the global model, but the performance was worse.

ALGORITHM-2

INPUT: $\mathbf{x}$, true labeling $\mathbf{y}^*$, current local model $P(\mathbf{h}|\mathbf{x})$
GOAL: Find a high confidence labeling $\mathbf{y}'$ that is
close to the true labeling $\mathbf{y}^*$

$\mathbf{h}^* \leftarrow \text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})$
$\mathbf{h}' \leftarrow$ single-link-clustering($\mathbf{h}^*$)
for each $h'_i \in \mathbf{h}'$
  if $h'_i = y^*_i$
    $y'_i \leftarrow h^*_i$
  else
    $y'_i \leftarrow y^*_i$
end for
return $\mathbf{y}'$

Figure 2: Algorithm to find a high confidence labeling $\mathbf{y}'$ that is close to the true labeling $\mathbf{y}^*$

We propose another approximation option for the E-step that is given by Figure 2. Intuitively, when the current local model misses positive coreference decisions, the first algorithm constructs a $\mathbf{y}'$ that is closest to $\mathbf{h}'$ for single-link clustering to recover the true labeling $\mathbf{y}^*$, while the second algorithm constructs a $\mathbf{y}'$ that is closer to $\mathbf{y}^*$ by preserving all of the missing positive coreference decisions. [6]

[Computation of M-step] Because $P(\mathbf{y}|\mathbf{h})$ is not parameterized, finding $\text{argmax}_\theta\ P(\mathbf{y}, \mathbf{h}|\mathbf{x})$ reduces to finding $\text{argmax}_\theta\ P(\mathbf{h}|\mathbf{x})$, which is standard CRF training. In order to speed up the training, we start convex optimization for CRFs using the parameter values $\theta^{(t-1)}$ from the previous M-step. For the very first iteration of EM, we start by setting $P(\mathbf{y}^*|\mathbf{x}) = 1$ for E-step, so that the first M-step will finds $\text{argmax}_\theta\ P(\mathbf{y}^*|\mathbf{x})$.

[6] In a pilot study, we found that ALGORITHM-2 performs slightly better than ALGORITHM-1. We also tried two other approximation options, but none performed as well as ALGORITHM-2. One of them removes the confusing sub-instances and has the effect of setting a uniform distribution on those sub-instances. The other computes the actual distribution on a subset of sub-instances. For brevity, we only present experimental results using ALGORITHM-2 in this paper.

[Inference on the test data] It is intractable to marginalize out $\mathbf{h}$ from $P(\mathbf{y}, \mathbf{h}|\mathbf{x})$. Therefore, similar to the Viterbi-training in the E-step, we approximate the distribution of $\mathbf{h}$ by $\text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{X})$.

3 Experiments–I

Data set: We evaluate our approach with two coreference data sets: MUC6 (MUC-6, 1995) and MPQA[7](Wiebe et al., 2005). For the MUC6 data set, we extract noun phrases (mentions) automatically, but for MPQA, we assume mentions for coreference resolution are given as in Stoyanov and Cardie (2006). For MUC6, we use the standard training/test data split. For MPQA, we use 150 documents for training, and 50 documents for testing.

Configuration: We follow Ng and Cardie (2002) for feature vector construction for each pair of mentions,[8] and Finley and Joachims (2005) for constructing a training/testing instance for each document: a training/testing instance consists of all pairs of mentions in a document. Then, a single pair of mentions is a *sub-instance*. We use the Mallet[9] implementation of CRFs, and set a Gaussian prior of 1.0 for all experiments. At each M-step, we train CRFs starting from the parameters from the previous M-step. We train CRFs up to 200 iterations, but because we start training CRFs from the previous parameters, the convergence from the second M-step becomes much faster. We apply up to 5 EM iterations, and choose best performing $\theta^{(t)}, 2 \leq t \leq 5$ based on the performance on the training data.[10]

Hypothesis: For the baseline (BASE) we employ the locally trained model for pairwise decisions without global inference. Clustering is applied only at test time, in order to make the assignment on the output variables coherent. We hypothesize that for the baseline, maximizing the likelihood for training will correlate more with the pairwise accuracy of the incoherent decisions before clustering than the pairwise accuracy of the coherent decisions after clustering. We also hypothesize that by performing structured local training (SLT), maximizing the likelihood will correlate more with the pairwise accuracy after clustering.

MUC6								
	after clustering				before clustering			
	e %	R %	P %	F %	e %	R %	P %	F %
BASE	1.50	**59.2**	56.2	**57.7**	1.18	38.0	85.6	52.6
SLT	**1.28**	49.8	**67.3**	57.2	1.35	26.4	84.3	40.2
MPQA								
	after clustering				before clustering			
	e %	R %	P %	F %	e %	R %	P %	F %
BASE	9.83	**75.8**	57.0	65.1	7.05	52.1	83.4	64.1
SLT	**6.39**	62.1	**80.6**	**70.2**	7.39	43.7	90.1	58.9

Table 1: Performance of Structured Local Training: SLT reduces error rate (e %) after applying single-link clustering.

Results: Experimental results are shown in Table 1. We report error rate (error rate $= 100 -$ accuracy) on the pairwise decisions (e %), and F1-score (F %) on the coreferent pairs.[11] For comparison, we show numbers from both after and before single-link clustering is applied. As hypothesized, the error rate of BASE increases after clustering, while the error rate of SLT decreases after clustering. Moreover, the error rate of SLT is considerably lower than that of BASE after clustering. However, the F1-score does not correlate with the error rate. That is, a lower error rate does not always lead to a higher F1-score, which motivates the *Biased Potential Functions* that we introduce in the next section. Notice that when we compare the precision/recall breakdown after clustering, SLT has higher precision and lower recall than BASE.

4 Biased Potential Functions

We introduce *biased potential functions* for training CRFs to empirically favor preferred evaluation measures for the learning task, such as F-score and MUC-score that have been considered hard for tradi-

[7] Available at http://nrrc.mitre.org/NRRC/publications.htm.

[8] In particular, our feature set corresponds to "All Features" in Ng and Cardie (2002), and we discretized numeric values.

[9] Available at http://mallet.cs.umass.edu.

[10] Selecting $\theta^{(t)}$ on a separate tuning data would be better, but the data for MUC6 in particular is very limited. Notice that we don't pick θ^1 when reporting the performance of SLT, because it is identical to the baseline.

[11] Error rate and F1-score on the coreferent pairs are not ideal measures for the quality of clustering, however, we show them here in order to contrast the effect of SLT. We present MUC-scores for the same experimental settings in Table 3.

tional likelihood-based methods to optimize for. Intuitively, biased potential functions emphasize those sub-components of an instance that can be of greater importance than the rest of an instance.

4.1 Definitions

The conditional probability of $P(\mathbf{y}|\mathbf{x})$[12] for CRFs is given by (Lafferty et al., 2001)

$$P(\mathbf{y}|\mathbf{x}) = \frac{\prod_i \phi(C_i, \mathbf{x})}{\sum_{\mathbf{y}} \prod_i \phi(C_i, \mathbf{x})}$$

where $\phi(C_i, \mathbf{x})$ is a potential function defined over each clique C_i. Potential functions are typically parameterized in an exponential form as follows.

$$\phi(C_i, \mathbf{x}) = \exp \sum_k \lambda_k f_k(C_i, \mathbf{x})$$

where λ_k are the parameters and $f_k(\cdot)$ are feature indicator functions. Because the Hammersley-Clifford theorem (1971) for undirected graphical models holds for any non-negative potential functions, we propose alternative potential functions as follows.

$$\psi(C_i, \mathbf{x}) = \begin{cases} \beta\phi(C_i, \mathbf{x}) & \text{if } \mu(C_i, \mathbf{x}) = true \\ \phi(C_i, \mathbf{x}) & \text{otherwise} \end{cases}$$

where β is a non-negative bias factor, and $\mu(C_i, \mathbf{x})$ is a predicate (or an indicator function) to check certain properties on $(C_i, \mathbf{x})$.[13] Examples of possible $\mu(\cdot)$ would be whether the true assignment for C_i in the training data contains certain class values, or whether the current observation indexed by C_i has particular characteristics. More specific details will be given in §4.2.

Training and testing with biased potential functions is mostly identical to the traditional log-linear formulations by $\phi(\cdot)$ as defined above, except for small and straightforward modifications to the computation of the likelihood and the derivative of the likelihood.

The key idea for biased potential functions is nothing new, as it is conceptually similar to instance weighting for problems with non-structured output (e.g. Aha and Goldstone (1992), Cardie et al. (1997)). However, biased potential functions differ technically in that they emphasize desired subcomponents without altering the i.i.d. assumption, and still weight each instance alike. Despite the conceptual simplicity, we are not aware of any previous work that explored biased potential functions for problems with structured output.

4.2 Applications to Coreference Resolution

[Bias on Coreferent Pairs] For coreference resolution, pairs that are coreferent are in a minority class[14], and biased potential functions can mitigate this skewed data problem, by amplifying the clique potentials that correspond to coreferent pairs. We define $\mu(y_i, x_i)$ to be true if and only if the true assignment for y_i in the training data is *'coreferent'*. Notice that $\mu(\cdot)$ does not depend on what particular value y_i might take, but only depends on the true value of y_i in the training data. For testing, $\mu(y_i, x_i)$ will be always false.[15]

[Bias on Closer Coreferent Pairs] For coreference resolution, we hypothesize that coreferent pairs for closer mentions have more significance, because they tend to have clearer linguistic clues to determine coreference. We further hypothesize that by emphasizing only close coreferent pairs, we can have our model favor the MUC score. For this, we define $\mu(y_i, x_i)$ to be true if and only if x_i is for a pair of mentions that are the closest coreferent pair.

5 Experiments–II

Data sets and configurations for experiments are identical to those used in §3.

Hypothesis: We hypothesize that using biased potential functions, maximizing the likelihood for training can correlate better with F1-score or MUC-score than the pairwise accuracy. In particular,

[12] For the local model described in Section 2, **y** should be replaced with **h**. We use **y** in this section however, as it is a more conventional notation in general.

[13] In our problem formulation, cliques are individual nodes, and potential functions are defined over the observations indexed by the current i only: i.e. $\phi(C_i, \mathbf{x}) = \phi(y_i, x_i)$, $\mu(C_i, \mathbf{x}) = \mu(y_i, x_i)$ and $\psi(C_i, \mathbf{x}) = \psi(y_i, x_i)$.

[14] Only 1.72% of the pairs are coreferent in the MUC6 data, and about 12% are coreferent in the MPQA data.

[15] Notice that $\mu(y_i, x_i)$ changes the surface of the likelihood for training, but does not affect the inference of finding the argmax in our local model. That is, $\text{argmax}_{y_i} \phi(y_i, x_i) = \text{argmax}_{y_i} \psi(y_i, x_i)$ (with y_i replaced with h_i).

MUC6							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	1.18	38.0	**85.6**	52.6	59.0	**75.8**	66.4
BASIC-P1$^{1.5}$	1.20	38.9	82.1	52.8	64.2	71.8	**67.8**
BASIC-P1$^{3.0}$	1.32	46.9	71.3	56.6	68.9	64.3	66.5
BASIC-P$a^{1.5}$	**1.15**	44.2	79.9	56.9	62.1	68.7	65.2
BASIC-P$a^{3.0}$	1.44	**52.5**	62.9	**57.2**	**70.9**	60.5	65.3
MPQA							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	**7.05**	52.1	**83.4**	64.1	75.6	**81.5**	**78.4**
BASIC-P1$^{1.5}$	7.18	54.6	79.6	64.8	77.7	76.5	77.1
BASIC-P1$^{3.0}$	7.22	59.9	75.4	66.8	83.3	71.7	77.1
BASIC-P$a^{1.5}$	7.65	59.7	72.2	65.4	79.8	73.2	76.4
BASIC-P$a^{3.0}$	8.22	**69.2**	65.1	**67.1**	**85.8**	67.8	75.7

Table 2: Performance of Biased Potential Functions: pairwise scores are taken before single-link-clustering is applied.

MUC6							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	1.50	59.2	56.2	57.7	59.0	75.8	66.4
SLT	1.28	49.8	67.3	57.2	56.3	**77.8**	65.3
SLT-P1$^{1.5}$	**1.19**	52.8	**70.6**	60.4	59.3	74.6	66.1
SLT-P1$^{3.0}$	1.42	63.5	57.9	**60.6**	67.5	70.7	69.1
SLT-P$a^{1.5}$	1.43	58.6	58.5	58.5*	64.0	73.6	68.5
SLT-P$a^{3.0}$	1.71	**65.2**	50.3	56.8	**70.5**	69.3	**69.9***
MPQA							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	9.83	75.8	57.0	65.1	75.6	81.5	78.4
SLT	**6.39**	62.1	**80.6**	70.2	69.1	**88.2**	77.5
SLT-P1$^{1.5}$	6.54	64.9	77.4	**70.6***	72.2	84.5	77.9*
SLT-P1$^{3.0}$	9.09	77.2	59.6	67.3	78.4	79.5	78.9
SLT-P$a^{1.5}$	6.74	65.2	75.7	70.1	72.4	87.2	**79.1**
SLT-P$a^{3.0}$	14.71	**78.2**	43.9	56.2	**80.5**	73.8	77.0

Table 3: Performance of Biased Potential Functions with Structured Local Training: All numbers are taken after single-link clustering.

we hypothesize that biasing on every coreferent pair will correlate more with F1-score, and biasing on close coreferent pairs will correlate more with MUC-score. In general, we expect that biasing on coreferent pairs will boost recall, potentially decreasing precision.

Results [BPF]: Experimental results for biased potential functions, without structured local training, are shown in Table 2. BASIC-P1$^{\beta}$ denotes local training with biased potential on the closest coreferent pairs with bias factor β, and BASIC-Pa^{β} denotes local training with biased potential on the all coreferent pairs with bias factor β, where $\beta = 1.5$ or 3.0. For brevity, we only show pairwise numbers before applying single-link-clustering.[16] As hypothesized, biased potential functions in general boost recall at the cost of precision. Also, for a fixed value of β, BASIC-P1$^{\beta}$ gives better MUC-F1 than BASIC-Pa^{β}, and BASIC-Pa^{β} gives better pairwise-F1 than BASIC-P1$^{\beta}$ for both data sets.

Results [SLT+BPF]: Experimental results that combine SLT and BPF are shown in Table 3. Similarly as before, SLT-Px^{β} denotes SLT with biased potential scheme Px, with bias factor β. For brevity, we only show numbers after applying single-link-clustering. Unlike the results shown in Table 2, for a fixed value of β, SLT-P1$^{\beta}$ correlates better with pairwise-F1, and SLT-Pa^{β} correlates better with MUC-F1. This indicates that when biased potential functions are used in conjunction with SLT, the effect of biased potential functions can be different from the case without SLT. Comparing F1-scores in Table 2 and Table 3, we see that the combination of biased potential functions with SLT improves performance in general. In particular, SLT-P1$^{3.0}$ and SLT-P$a^{1.5}$ consistently improve performance over BASE on both data sets, for both pairwise-F1 and MUC-F1. We present performance scores for all variations of configurations for reference, but we also mark the particular configuration SLT-Px^{β} (by '*' on F1-scores) that is chosen when selecting the configuration based on the performance on the training data for each performance measure. To conclude, structured local training with biased potential functions bring a substantial improvement for MUC-F1 score, from 66.4% to 69.9% for MUC6 data set. For pairwise-F1, the performance increase from 57.7% to 58.5% for MUC6, and from 65.1% to 70.6% for MPQA.[17]

[16] This is because we showed in §3 that basic local training does not correlate well with pairwise scores after clustering, and in order to see the direct effect of biased potential functions, we examine pairwise numbers before clustering.

[17] Performance on the MPQA data for MUC-F1 is slightly decreased from 78.4% to 77.9%. Note the MUC scores for the

6 Related Work

Structured local training is motivated by recent research that has shown that reducing the discrepancy between the training model and testing model can improve the performance without incurring the heavy computational overhead of full-blown global inference-based training. [18] (e.g. Cohen and Carvalho (2005), Sutton and McCallum (2005a), Sutton and McCallum (2005b)). Our work differs in that (1) we use hidden variables to capture the interactions between local inference and global inference, (2) we present an application to coreference resolution, while previous work has shown applications for variants of sequence tagging. McCallum and Wellner (2004) showed a global training approach with CRFs for coreference resolution, but they used the voted perceptron algorithm for training, which no longer maximizes the likelihood. In addition, they assume that all and only those noun phrases involved in coreference resolution are given.

The performance of our system on MUC6 data set is comparable to previously reported systems. Using the same feature set, Ng and Cardie (2002) reports 64.5% of MUC-score, while our system achieved 69.9%. Ng and Cardie (2002) reports 70.4% of MUC-score using hand-selected features. With an additional feature selection or feature induction step, the performance of our system might further improve. McCallum and Wellner (2004) reports 73.42% of MUC-score on MUC6 data set, but their experiments assumed perfect identification of all and only those noun phrases involved in a coreference relation, thus substantially simplifying the task.

7 Conclusion

We present a novel training procedure, *structured local training*, that maximizes likelihood while exploiting the benefits of global inference during training. This is achieved by incorporating hidden variables to capture the interactions between local inference and global inference. In addition, we introduce *biased potential functions* that allow CRFs to empirically favor performance measures such as F1-score or MUC-score. We focused on the application of coreference resolution in this paper, but the key ideas of our approaches can be extended to other applications, and other machine learning techniques motivated by Markov networks.

Acknowledgments We thank the reviewers as well as Eric Breck and Ves Stoyanov for their many helpful comments. This work was supported by the Advanced Research and Development Activity (ARDA), by NSF Grants BCS-0624277, IIS-0535099, and IIS-0208028, and by gifts from Google and the Xerox Foundation.

MPQA baseline are already quite high to begin with.

[18] The computational cost for SLT in our experiments were about twice of the cost for the local training of the baseline. This is the case because M-step converges very fast from the second EM iteration, by initializing CRFs using parameters from the previous M-step. Biased potential functions hardly adds extra computational cost. In practice, BPFs reduce training time substantially: we observed that the higher the bias is, the quicker CRFs converge.

References

D.W. Aha and R.L. Goldstone. 1992. Concept learning and flexible weighting. In *Proc. of the Fourteenth Annual Conference of the Cognitive Science Society*.

A. Berger, S.D. Pietra, V.D. Pietra 1996. A Maximum Entropy Approach to Natural Language Processing. In *Computational Linguistics*,22.

C. Cardie and N. Howe. 1997. Improving Minority Class Prediction Using Case-Specific Feature Weights. In *ICML*.

W.W. Cohen and V. Carvalho. 2005. Stacked Sequential Learning. In *IJCAI*.

M. Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *EMNLP*.

A.P. Dempster, N. M. Laird and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. In *Journal of the Loyal Statistical Society*, B.39.

J. Finkel, T. Grenager and C. D. Manning. 2005. Incorporating Non-local Information Into Information Extraction Systems By Gibbs Sampling. In *ACL*.

T. Finley and T. Joachims. 2005. Supervised Clustering with Support Vector Machines. In *ICML*.

J. Hammersley and P. Clifford. 1971. Markov fields on finite graphs and lattices. Unpublished manuscript.

J. Lafferty, A. McCallum and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*.

A. McCallum and B. Wellner. 2004. Conditional Models of Identity Uncertainty with Application to Noun Coreference. In *NIPS*.

MUC-6 1995. In Proc. of the Sixth Message Understanding Conference (MUC-6) Morgan Kaufmann.

R. M. Neal and G. E. Hinton. 1998. A view of the EM algorithm that justies incremental, sparse, and other variants. In *Learning in Graphical Models*, Kluwer.

V. Ng and C. Cardie. 2002. Improving Machine Learning Approaches to Coreference Resolution. In *ACL*.

V. Punyakanok, D. Roth, W. Yih, and D. Zimak 2005. Learning and Inference over Constrained Output. In *IJCAI*.

D. Roth and W. Yih. 2005. Integer Linear Programming Inference for Conditional Random Fields. In *ICML*.

V. Stoyanov and C. Cardie. 2006. Partially Supervised Coreference Resolution for Opinion Summarization through Structured Rule Learning. In *EMNLP*.

C. Sutton and A. McCallum. 2005. Fast, Piecewise Training for Discriminative Finite-state and Parsing Models. In *Technical Report IR-403, University of Massachusetts*.

C. Sutton and A. McCallum. 2005. Piecewise Training for Undirected Models. In *UAI*.

B. Wellner, A. McCallum, F. Peng and M. Hay. 2004. An Integrated, Conditional Model of Information Extraction and Coreference with Application to Citation Matching. In *UAI*.

J. Wiebe and T. Wilson and C. Cardie 2005. Annotating Expressions of Opinions and Emotions in Language. In *Language Resources and Evaluation, volume 39, issue 2-3*.