Unified Expectation Maximization

Rajhans Samdani
University of Illinois
rsamdan2@illinois.edu

Ming-Wei Chang
Microsoft Research
minchang@microsoft.com

Dan Roth
University of Illinois
danr@illinois.edu

Abstract

We present a general framework containing a graded spectrum of Expectation Maximization (EM) algorithms called Unified Expectation Maximization (UEM.) UEM is parameterized by a single parameter and covers existing algorithms like standard EM and hard EM, constrained versions of EM such as Constraint-Driven Learning (Chang et al., 2007) and Posterior Regularization (Ganchev et al., 2010), along with a range of new EM algorithms. For the constrained inference step in UEM we present an efficient dual projected gradient ascent algorithm which generalizes several dual decomposition and Lagrange relaxation algorithms popularized recently in the NLP literature (Ganchev et al., 2008; Koo et al., 2010; Rush and Collins, 2011). UEM is as efficient and easy to implement as standard EM. Furthermore, experiments on POS tagging, information extraction, and word-alignment show that often the best performing algorithm in the UEM family is a new algorithm that wasn't available earlier, exhibiting the benefits of the UEM framework.

1 Introduction

Expectation Maximization (EM) (Dempster et al., 1977) is inarguably the most widely used algorithm for unsupervised and semi-supervised learning. Many successful applications of unsupervised and semi-supervised learning in NLP use EM including text classification (McCallum et al., 1998; Nigam et al., 2000), machine translation (Brown et al., 1993), and parsing (Klein and Manning, 2004). Recently, EM algorithms which incorporate constraints on structured output spaces have been proposed (Chang et al., 2007; Ganchev et al., 2010).

Several variations of EM (e.g. hard EM) exist in the literature and choosing a suitable variation is often very task-specific. Some works have shown that for certain tasks, hard EM is more suitable than regular EM (Spitkovsky et al., 2010). The same issue continues in the presence of constraints where Posterior Regularization (PR) (Ganchev et al., 2010) corresponds to EM while Constraint-Driven Learning (CoDL)[1] (Chang et al., 2007) corresponds to hard EM. The problem of choosing between EM and hard EM (or between PR and CoDL) remains elusive, along with the possibility of simple and better alternatives, to practitioners. Unfortunately, little study has been done to understand the relationships between these variations in the NLP community.

In this paper, we approach various EM-based techniques from a novel perspective. We believe that "EM or Hard-EM?" and "PR or CoDL?" are not the right questions to ask. Instead, we present a unified framework for EM, Unified EM (UEM), that covers many EM variations including the constrained cases along with a continuum of new ones. UEM allows us to compare and investigate the properties of EM in a systematic way and helps find better alternatives.

The contributions of this paper are as follows:

1. We propose a general framework called Unified Expectation Maximization (UEM) that presents a continuous spectrum of EM algorithms parameterized by a simple temperature-like tuning parameter. The framework covers both constrained and unconstrained EM algorithms. UEM thus connects EM, hard EM, PR, and CoDL so that the relation between different algorithms can be better understood. It also enables us to find new EM algorithms.
2. To solve UEM (with constraints), we propose

[1]To be more precise, (Chang et al., 2007) mentioned using hard constraints as well as soft constraints in EM. In this paper, we refer to CoDL only as the EM framework with hard constraints.

a dual projected subgradient ascent algorithm that generalizes several dual decomposition and Lagrange relaxation algorithms (Bertsekas, 1999) introduced recently in NLP (Ganchev et al., 2008; Rush and Collins, 2011).

3. We provide a way to implement a family of EM algorithms and choose the appropriate one, given the data and problem setting, rather than a single EM variation. We conduct experiments on unsupervised POS tagging, unsupervised word-alignment, and semi-supervised information extraction and show that choosing the right UEM variation outperforms existing EM algorithms by a significant margin.

2 Preliminaries

Let $\mathbf{x}$ denote an input or observed features and $\mathbf{h}$ be a discrete output variable to be predicted from a finite set of possible outputs $\mathcal{H}(\mathbf{x})$. Let $P_\theta(\mathbf{x}, \mathbf{h})$ be a probability distribution over $(\mathbf{x}, \mathbf{h})$ parameterized by θ. Let $P_\theta(\mathbf{h}|\mathbf{x})$ refer to the conditional probability of $\mathbf{h}$ given $\mathbf{x}$. For instance, in part-of-speech tagging, $\mathbf{x}$ is a sentence, $\mathbf{h}$ the corresponding POS tags, and θ could be an HMM model; in word-alignment, $\mathbf{x}$ can be an English-French sentence pair, $\mathbf{h}$ the word alignment between the sentences, and θ the probabilistic alignment model. Let $\delta(\mathbf{h} = \mathbf{h}')$ be the Kronecker-Delta distribution centered at $\mathbf{h}'$, i.e., it puts a probability of 1 at $\mathbf{h}'$ and 0 elsewhere.

In the rest of this section, we review EM and constraints-based learning with EM.

2.1 EM Algorithm

To obtain the parameter θ in an unsupervised way, one maximizes log-likelihood of the observed data:

$$\mathcal{L}(\theta) = \log P_\theta(\mathbf{x}) = \log \sum_{\mathbf{h} \in \mathcal{H}(\mathbf{x})} P_\theta(\mathbf{x}, \mathbf{h}) \quad . \tag{1}$$

EM (Dempster et al., 1977) is the most common technique for learning θ, which maximizes a tight lower bound on $\mathcal{L}(\theta)$. While there are a few different styles of expressing EM, following the style of (Neal and Hinton, 1998), we define

$$F(\theta, q) = \mathcal{L}(\theta) - KL(q, P_\theta(\mathbf{h}|\mathbf{x})), \tag{2}$$

where q is a posterior distribution over $\mathcal{H}(\mathbf{x})$ and $KL(p_1, p_2)$ is the KL divergence between two distributions p_1 and p_2. Given this formulation, EM can be shown to maximize F via block coordinate ascent alternating over q (E-step) and θ (M-step) (Neal and Hinton, 1998). In particular, the E-step for EM can be written as

$$q = \arg\min_{q' \in \mathcal{Q}} KL(q', P_\theta(\mathbf{h}|\mathbf{x})) \quad , \tag{3}$$

where $\mathcal{Q}$ is the space of all distributions. While EM produces a distribution in the E-step, hard EM is thought of as producing a single output given by

$$\mathbf{h}^* = \arg\max_{\mathbf{h} \in \mathcal{H}(\mathbf{x})} P_\theta(\mathbf{h}|\mathbf{x}) \quad . \tag{4}$$

However, one can also think of hard EM as producing a distribution given by $q = \delta(\mathbf{h} = \mathbf{h}^*)$. In this paper, we pursue this distributional view of both EM and hard EM and show its benefits.

EM for Discriminative Models EM-like algorithms can also be used in discriminative settings (Bellare et al., 2009; Ganchev et al., 2010) specifically for semi-supervised learning (SSL.) Given some labeled and unlabeled data, such algorithms maximize a modified $F(\theta, q)$ function:

$$F(\theta, q) = L_c(\theta) - c_1\|\theta\|^2 - c_2 KL(q, P_\theta(\mathbf{h}|\mathbf{x})) \quad , \tag{5}$$

where, q, as before, is a probability distribution over $\mathcal{H}(\mathbf{x})$, $L_c(\theta)$ is the conditional log-likelihood of the labels given the features for the labeled data, and c_1 and c_2 are constants specified by the user; the KL divergence is measured only over the unlabeled data.

The EM algorithm in this case has the same E-step as unsupervised EM, but the M-step is different. The M-step is similar to supervised learning as it finds θ by maximizing a regularized conditional likelihood of the data w.r.t. the labels — true labels are used for labeled data and "soft" pseudo labels based on q are used for unlabeled data.

2.2 Constraints in EM

It has become a common practice in the NLP community to use constraints on output variables to guide inference. Few of many examples include type constraints between relations and entities (Roth and Yih, 2004), sentential and modifier constraints during sentence compression (Clarke and Lapata, 2006), and agreement constraints between word-alignment directions (Ganchev et al., 2008) or various parsing models (Koo et al., 2010). In the context of EM, constraints can be imposed on the posterior probabilities, q, to guide the learning procedure (Chang et al., 2007; Ganchev et al., 2010).

In this paper, we focus on linear constraints over $\mathbf{h}$ (potentially non-linear over $\mathbf{x}$.) This is a very general formulation as it is known that all Boolean constraints can be transformed into sets of linear constraints over binary variables (Roth and Yih, 2007). Assume that we have m *linear* constraints on outputs where the k^{th} constraint can be written as

$$\mathbf{u_k}^T\mathbf{h} \leq b_k \ .$$

Defining a matrix U as $\mathbf{U}^T = \begin{bmatrix}\mathbf{u_1}^T & \dots & \mathbf{u_m}^T\end{bmatrix}$ and a vector $\mathbf{b}$ as $\mathbf{b}^T = [b_1, \dots, b_m]$, we write down the set of all *feasible*[2] structures as

$$\{\mathbf{h} \mid \mathbf{h} \in \mathcal{H}(\mathbf{x}), \mathbf{U}\mathbf{h} \leq \mathbf{b}\} \ .$$

Constraint-Driven Learning (CoDL) (Chang et al., 2007) augments the E-step of hard EM (4) by imposing these constraints on the outputs.

Constraints on structures can be relaxed to *expectation constraints* by requiring the distribution q to satisfy them only in expectation. Define expectation w.r.t. a distribution q over $\mathcal{H}(\mathbf{x})$ as $E_q[\mathbf{U}\mathbf{h}] = \sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} q(\mathbf{h})\mathbf{U}\mathbf{h}$. In the expectation constraints setting, q is required to satisfy:

$$E_q[\mathbf{U}\mathbf{h}] \leq \mathbf{b} \ .$$

The space of distributions $\mathcal{Q}$ can be modified as:

$$\mathcal{Q} = \{q \mid q(\mathbf{h}) \geq 0, E_q[\mathbf{U}\mathbf{h}] \leq \mathbf{b}, \sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} q(\mathbf{h}) = 1\}.$$

Augmenting these constraints into the E-step of EM (3), gives the Posterior Regularization (PR) framework (Ganchev et al., 2010). In this paper, we adopt the expectation constraint setting. Later, we show that UEM naturally includes and generalizes both PR and CoDL.

3 Unified Expectation Maximization

We now present the Unified Expectation Maximization (UEM) framework which captures a continuum of (constrained and unconstrained) EM algorithms including EM and hard EM by modulating the entropy of the posterior. A key observation underlying the development of UEM is that hard EM (or CoDL) finds a distribution with zero entropy while EM (or PR) finds a distribution with the same entropy as P_θ (or close to it). Specifically, we modify the objective of the E-step of EM (3) as

$$q = \arg\min_{q'\in\mathcal{Q}} KL(q', P_\theta(\mathbf{h}|\mathbf{x}); \gamma) \ , \qquad (6)$$

where $KL(q, p; \gamma)$ is a modified KL divergence:

$$KL(q,p;\gamma) = \sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} \gamma q(\mathbf{h}) \log q(\mathbf{h}) - q(\mathbf{h}) \log p(\mathbf{h}). \quad (7)$$

In other words, UEM projects $P_\theta(\mathbf{h}|\mathbf{x})$ on the space of feasible distributions $\mathcal{Q}$ w.r.t. a metric[3] $KL(\cdot,\cdot;\gamma)$ to obtain the posterior q. By simply varying γ, UEM changes the metric of projection and obtains different variations of EM including EM (PR, in the presence of constraints) and hard EM (CoDL.) The M-step for UEM is exactly the same as EM (or discriminative EM.)

The UEM Algorithm: Alg. 1 shows the UEM algorithm for both the generative (G) and the discriminative (D) case. We refer to the UEM algorithm with parameter γ as UEM$_\gamma$.

Algorithm 1 The UEM algorithm for both the generative (G) and discriminative (D) cases.

```
Initialize $\theta^0$
for $t = 0, \dots, T$ do
    UEM E-step:
    $q^{t+1} \leftarrow \arg\min_{q\in\mathcal{Q}} KL(q, P_{\theta^t}(\mathbf{h}|\mathbf{x}); \gamma)$
    UEM M-step:
    G: $\theta^{t+1} = \arg\max_\theta E_{q^{t+1}}[\log P_\theta(\mathbf{x},\mathbf{h})]$
    D: $\theta^{t+1} = \arg\max_\theta E_{q^{t+1}}[\log P_\theta(\mathbf{h}|\mathbf{x})] - c_1\|\theta\|^2$
end for
```

3.1 Relationship between UEM and Other EM Algorithms

The relation between unconstrained versions of EM has been mentioned before (Ueda and Nakano, 1998; Smith and Eisner, 2004). We show that the relationship takes novel aspects in the presence of constraints. In order to better understand different UEM variations, we write the UEM E-step (6) explicitly as an optimization problem:

[2]Note that this set is a finite set of discrete variables not to be confused with a polytope. Polytopes are also specified as $\{\mathbf{z}|\mathbf{A}\mathbf{z} \leq \mathbf{d}\}$ but are over real variables whereas $\mathbf{h}$ is discrete.

[3]The term 'metric' is used very loosely. $KL(\cdot,\cdot;\gamma)$ does not satisfy the mathematical properties of a metric.

Framework	$\gamma=-\infty$	$\gamma=0$	$\gamma\in(0,1)$	$\gamma=1$	$\gamma=\infty\to 1$
Unconstrained	Hard EM	Hard EM	(NEW) UEM$_\gamma$	Standard EM	Deterministic Annealing EM
Constrained	CoDL (Chang et al., 2007)	(NEW) EM with Lin. Prog.	(NEW) constrained UEM$_\gamma$	PR (Ganchev et al., 2010)	

Table 1: Summary of different UEM algorithms. The entries marked with "(NEW)" have not been proposed before. Eq. (8) is the objective function for all the EM frameworks listed in this table. Note that, in the absence of constraints, $\gamma \in (-\infty, 0]$ corresponds to hard EM (Sec. 3.1.1.) Please see Sec. 3.1 for a detailed explanation.

$$\begin{aligned} \min_q \quad & \sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} \gamma q(\mathbf{h}) \log q(\mathbf{h}) - q(\mathbf{h}) \log P_\theta(\mathbf{h}|\mathbf{x}) \quad (8) \\ \text{s.t.} \quad & E_q[\mathbf{U}\mathbf{h}] \le \mathbf{b}, \\ & q(\mathbf{h}) \ge 0, \forall \mathbf{h} \in \mathcal{H}(\mathbf{x}), \\ & \textstyle\sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} q(\mathbf{h}) = 1 \ . \end{aligned}$$

We discuss below, both the constrained and the unconstrained cases. Tab. 1 summarizes different EM algorithms in the UEM family.

3.1.1 UEM Without Constraints

The E-step in this case, computes a q obeying only the simplex constraints: $\sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} q(\mathbf{h}) = 1$. For $\gamma = 1$, UEM minimizes $KL(q, P_\theta(\mathbf{h}|\mathbf{x}); 1)$ which is the same as minimizing $KL(q, P_\theta(\mathbf{h}|\mathbf{x}))$ as in the standard EM (3). For $\gamma = 0$, UEM is solving $\arg\min_{q\in\mathcal{Q}} \sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} -q(\mathbf{h}) \log P_\theta(\mathbf{h}|\mathbf{x})$ which is a linear programming (LP) problem. Due to the unimodularity of the simplex constraints (Schrijver, 1986), this LP outputs an integral $q = \delta\left(\mathbf{h} = \arg\max_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} P_\theta(\mathbf{h}|x)\right)$ which is the same as hard EM (4). It has already been noted in the literature (Kearns et al., 1997; Smith and Eisner, 2004; Hofmann, 2001) that this formulation (corresponding to our $\gamma = 0$) is the same as hard EM. In fact, for $\gamma \le 0$, UEM stays the same as hard EM because of negative penalty on the entropy. The range $\gamma \in (0, 1)$ has not been discussed in the literature, to the best of our knowledge. In Sec. 5, we show the impact of using UEM$_\gamma$for $\gamma \in \{0, 1\}$. Lastly, the range of γ from ∞ to 1 has been used in deterministic annealing for EM (Rose, 1998; Ueda and Nakano, 1998; Hofmann, 2001). However, the focus of deterministic annealing is solely to solve the standard EM while avoiding local maxima problems.

3.1.2 UEM With Constraints

UEM and Posterior Regularization ($\gamma = 1$) For $\gamma = 1$, UEM solves $\arg\min_{q\in\mathcal{Q}} KL\left(q, P_\theta(\mathbf{h}|\mathbf{x})\right)$ which is the same as Posterior Regularization (Ganchev et al., 2010).

UEM and CoDL ($\gamma = -\infty$) When $\gamma \to -\infty$ then due to an infinite penalty on the entropy of the posterior, the entropy must become zero. Thus, now the E-step, as expressed by Eq. (8), can be written as $q = \delta(\mathbf{h} = \mathbf{h}^*)$ where $\mathbf{h}^*$ is obtained as

$$\begin{aligned} \arg\max_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} \quad & \log P_\theta(\mathbf{h}|\mathbf{x}) \qquad (9) \\ \text{s.t.} \quad & \mathbf{U}\mathbf{h} \le \mathbf{b} \ , \end{aligned}$$

which is the same as CoDL. This combinatorial maximization can be solved using the Viterbi algorithm in some cases or, in general, using Integer Linear Programming (ILP.)

3.2 UEM with $\gamma \in [0, 1]$

Tab. 1 lists different EM variations and their associated values γ. This paper focuses on values of γ between 0 and 1 for the following reasons. First, the E-step (8) is non-convex for $\gamma < 0$ and hence computationally expensive; e.g., hard EM (i.e. $\gamma = -\infty$) requires ILP inference. For $\gamma \ge 0$, (8) is a convex optimization problem which can be solved exactly and efficiently. Second, for $\gamma = 0$, the E-step solves

$$\begin{aligned} \max_q \quad & \textstyle\sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} q(\mathbf{h}) \log P_\theta(\mathbf{h}|\mathbf{x}) \qquad (10) \\ \text{s.t.} \quad & E_q[\mathbf{U}\mathbf{h}] \le \mathbf{b}, \\ & q(\mathbf{h}) \ge 0, \forall \mathbf{h} \in \mathcal{H}(\mathbf{x}), \\ & \textstyle\sum_{\mathbf{h}\in\mathcal{H}(\mathbf{x})} q(\mathbf{h}) = 1 \ , \end{aligned}$$

which is an **LP-relaxation** of hard EM (Eq. (4) and (9)). LP relaxations often provide a decent proxy to ILP (Roth and Yih, 2004; Martins et al., 2009). Third, $\gamma \in [0, 1]$ covers standard EM/PR.

3.2.1 Discussion: Role of γ

The modified KL divergence can be related to standard KL divergence as $KL(q, P_\theta(\mathbf{h}|\mathbf{x}); \gamma) =$

$KL(q, P_\theta(\mathbf{y}|\mathbf{x})) + (1-\gamma)H(q)$ — UEM (6) minimizes the former during the E-step, while Standard EM (3) minimizes the latter. The additional term $(1-\gamma)H(q)$ is essentially an entropic prior on the posterior distribution q which can be used to regularize the entropy as desired.

For $\gamma < 1$, the regularization term penalizes the entropy of the posterior thus reducing the probability mass on the tail of the distribution. This is significant, for instance, in unsupervised structured prediction where the tail can carry a substantial amount of probability mass as the output space is massive. This notion aligns with the observation of (Spitkovsky et al., 2010) who criticize EM for frittering away too much probability mass on unimportant outputs while showing that hard EM does much better in PCFG parsing. In particular, they empirically show that when initialized with a "good" set of parameters obtained by supervised learning, EM drifts away (thus losing accuracy) much farther than hard-EM.

4 Solving Constrained E-step with Lagrangian Dual

In this section, we discuss how to solve the E-step (8) for UEM. It is a non-convex problem for $\gamma < 0$; however, for $\gamma = -\infty$ (CoDL) one can use ILP solvers. We focus here on solving the E-step for $\gamma \geq 0$ for which it is a convex optimization problem, and use a Lagrange relaxation algorithm (Bertsekas, 1999). Our contributions are two fold:

- We describe an algorithm for UEM with constraints that is as easy to implement as PR or CoDL. Existing code for constrained EM (PR or CoDL) can be easily extended to run UEM.
- We solve the E-step (8) using a Lagrangian dual-based algorithm which performs projected subgradient-ascent on dual variables. Our algorithm covers Lagrange relaxation and dual decomposition techniques (Bertsekas, 1999) which were recently popularized in NLP (Rush and Collins, 2011; Rush et al., 2010; Koo et al., 2010). Not only do we extend the algorithmic framework to a continuum of algorithms, we also allow, unlike the aforementioned works, general inequality constraints over the output variables. Furthermore, we establish new and interesting connections between existing constrained inference techniques.

4.1 Projected Subgradient Ascent with Lagrangian Dual

We provide below a high-level view of our algorithm, omitting the technical derivations due to lack of space. To solve the E-step (8), we introduce dual variables λ — one for each expectation constraint in $\mathcal{Q}$. The subgradient $\nabla\lambda$ of the dual of Eq. (8) w.r.t. λ is given by

$$\nabla\lambda \propto E_q[\mathbf{U}\mathbf{h}] - \mathbf{b} \ . \tag{11}$$

For $\gamma > 0$, the primal variable q can be written in terms of λ as

$$q(\mathbf{h}) \propto P_{\theta^t}(\mathbf{h}|\mathbf{x})^{\frac{1}{\gamma}} e^{-\frac{\lambda^T \mathbf{U}\mathbf{h}}{\gamma}} \ . \tag{12}$$

For $\gamma = 0$, the q above is not well defined and so we take the limit $\gamma \to 0$ in (12) and since l_p norm approaches the max-norm as $p \to \infty$, this yields

$$q(\mathbf{h}) = \delta(\mathbf{h} = \arg\max_{\mathbf{h}' \in \mathcal{H}(\mathbf{x})} P_\theta(\mathbf{h}'|\mathbf{x}) e^{-\lambda^T \mathbf{U}\mathbf{h}'}). \tag{13}$$

We combine both the ideas by setting $q(\mathbf{h}) = G(\mathbf{h}, P_{\theta^t}(\cdot|\mathbf{x}), \lambda^T\mathbf{U}, \gamma)$ where

$$G(\mathbf{h}, P, \mathbf{v}, \gamma) = \begin{cases} \frac{P(\mathbf{h})^{\frac{1}{\gamma}} e^{-\frac{\mathbf{v}\mathbf{h}}{\gamma}}}{\sum_{\mathbf{h}'} P(\mathbf{h}')^{\frac{1}{\gamma}} e^{-\frac{\mathbf{v}\mathbf{h}'}{\gamma}}} & \gamma > 0 \ , \\ \delta(\mathbf{h}{=}\arg\max_{\mathbf{h}' \in \mathcal{H}(\mathbf{x})} P(\mathbf{h}') e^{-\mathbf{v}\mathbf{h}'}) & \gamma = 0 \ . \end{cases} \tag{14}$$

Alg. 2 shows the overall optimization scheme. The dual variables for inequality constraints are restricted to be positive and hence after a gradient update, negative dual variables are projected to 0.

Note that for $\gamma = 0$, our algorithm is a Lagrange relaxation algorithm for approximately solving the E-step for CoDL (which uses exact $\arg\max$ inference). Lagrange relaxation has been recently shown to provide exact and optimal results in a large number of cases (Rush and Collins, 2011). This shows that our range of algorithms is very broad — it includes PR and a good approximation to CoDL.

Overall, the required optimization (8) can be solved efficiently if the expected value computation in the dual gradient (Eq. (11)) w.r.t. the posterior q in the primal (Eq (14)) can be performed efficiently. In cases where we can enumerate the possible outputs $\mathbf{h}$ efficiently, e.g. multi-class classification, we

Algorithm 2 Solving E-step of UEM$_\gamma$ for $\gamma \geq 0$.

```
1: Initialize and normalize $q$; initialize $\lambda = \mathbf{0}$.
2: for $t = 0, \ldots, R$ or until convergence do
3:     $\lambda \leftarrow \max\left(\lambda + \eta_t\left(E_q[\mathbf{U}\mathbf{h}] - \mathbf{b}\right), 0\right)$
4:     $q(\mathbf{h}) = G(\mathbf{h}, P_{\theta^t}(\cdot|\mathbf{x}), \lambda^T\mathbf{U}, \gamma)$
5: end for
```

can compute the posterior probability q explicitly using the dual variables. In cases where the output space is structured and exponential in size, e.g. word alignment, we can optimize (8) efficiently if the constraints and the model $P_\theta(\mathbf{h}|\mathbf{x})$ decompose in the same way. To elucidate, we give a more concrete example in the next section.

4.2 Projected Subgradient based Dual Decomposition Algorithm

Solving the inference (8) using Lagrangian dual can often help us decompose the problem into components and handle complex constraints in the dual space as we show in this section. Suppose our task is to predict two output variables $\mathbf{h}^1$ and $\mathbf{h}^2$ coupled via linear constraints. Specifically, they obey $\mathbf{U_e}\mathbf{h}^1 = \mathbf{U_e}\mathbf{h}^2$ (agreement constraints) and $\mathbf{U_i}\mathbf{h}^1 \leq \mathbf{U_i}\mathbf{h}^2$ (inequality constraints)[4] for given matrices $\mathbf{U_e}$ and $\mathbf{U_i}$. Let their respective probabilistic models be $P^1_{\theta_1}$ and $P^2_{\theta_2}$. The E-step (8) can be written as

$$\begin{aligned} \arg\min_{q_1,q_2} \quad & A(q_1, q_2; \gamma) \qquad (15)\\ s.t. \quad & E_{q_1}[\mathbf{U_e}\mathbf{h}^1] = E_{q_2}[\mathbf{U_e}\mathbf{h}^2] \\ & E_{q_1}[\mathbf{U_i}\mathbf{h}^1] \leq E_{q_2}[\mathbf{U_i}\mathbf{h}^2] \ , \end{aligned}$$

where $A(q_1, q_2; \gamma) = KL(q_1(\mathbf{h}^1), P^1_{\theta_1}(\mathbf{h}^1|\mathbf{x}); \gamma) + KL(q_2(\mathbf{h}^2), P^2_{\theta_2}(\mathbf{h}^2|\mathbf{x}); \gamma)$.

The application of Alg. 2 results in a dual decomposition scheme which is described in Alg. 3.

Note that in the absence of inequality constraints and for $\gamma = 0$, our algorithm reduces to a simpler dual decomposition algorithm with agreement constraints described in (Rush et al., 2010; Koo et al., 2010). For $\gamma = 1$ with agreement constraints, our algorithm specializes to an earlier proposed technique by (Ganchev et al., 2008). Thus our algorithm puts these dual decomposition techniques with

Algorithm 3 Projected Subgradient-based Lagrange Relaxation Algorithm that optimizes Eq. (15)

```
1: Input: Two distributions $P^1_{\theta_1}$ and $P^2_{\theta_2}$.
2: Output: Output distributions $q_1$ and $q_2$ in (15)
3: Define $\lambda^T = \begin{bmatrix}\lambda_\mathbf{e}^T & \lambda_\mathbf{i}^T\end{bmatrix}$ and $\mathbf{U}^T = \begin{bmatrix}\mathbf{U_e}^T & \mathbf{U_i}^T\end{bmatrix}$
4: $\lambda \leftarrow 0$
5: for $t = 0, \ldots, R$ or until convergence do
6:     $q_1(\mathbf{h}^1) \leftarrow G(\mathbf{h}^1, P^1_{\theta_1}(\cdot|\mathbf{x}), \lambda^T\mathbf{U}, \gamma)$
7:     $q_2(\mathbf{h}^2) \leftarrow G(\mathbf{h}^2, P^2_{\theta_2}(\cdot|\mathbf{x}), -\lambda^T\mathbf{U}, \gamma)$
8:     $\lambda_\mathbf{e} \leftarrow \lambda_\mathbf{e} + \eta_t(-E_{q_1}[\mathbf{U_e}\mathbf{h}^1] + E_{q_2}[\mathbf{U_e}\mathbf{h}^2])$
9:     $\lambda_\mathbf{i} \leftarrow \lambda_\mathbf{i} + \eta_t(-E_{q_1}[\mathbf{U_i}\mathbf{h}^1] + E_{q_2}[\mathbf{U_i}\mathbf{h}^2])$
10:    $\lambda_\mathbf{i} \leftarrow \max(\lambda_\mathbf{i}, 0)$ {Projection step}
11: end for
12: return $(q_1, q_2)$
```

agreement constraints on the same spectrum. Moreover, dual-decomposition is just a special case of Lagrangian dual-based techniques. Hence Alg. 2 is more broadly applicable (see Sec. 5). Lines 6-9 show that the required computation is decomposed over each sub-component.

Thus if computing the posterior and expected values of linear functions over each subcomponent is easy, then the algorithm works efficiently. Consider the case when constraints decompose linearly over $\mathbf{h}$ and each component is modeled as an HMM with θ_S as the initial state distribution, θ_E as emmision probabilities, and θ_T as transition probabilities. An instance of this is word alignment over language pair (S, T) modeled using an HMM augmented with agreement constraints which constrain alignment probabilities in one direction (P_{θ_1}: from S to T) to agree with the alignment probabilities in the other direction (P_{θ_2}: from T to S.) The agreement constraints are linear over the alignments, $\mathbf{h}$.

Now, the HMM probability is given by $P_\theta(\mathbf{h}|\mathbf{x}) = \theta_S(\mathbf{h}_0)\prod_i \theta_E(\mathbf{x}_i|\mathbf{h}_i)\theta_T(\mathbf{h}_{i+1}|\mathbf{h}_i)$ where $\mathbf{v}_i$ denotes the i^{th} component of a vector $\mathbf{v}$. For $\gamma > 0$, the resulting q (14) can be expressed using a vector μ =+/-$\lambda^T\mathbf{U}$ (see lines 6-7) as

$$\begin{aligned} q(\mathbf{h}) &\propto \left(\theta_S(\mathbf{h}_0)\prod_i \theta_E(\mathbf{x}_i|\mathbf{h}_i)\theta_T(\mathbf{h}_{i+1}|\mathbf{h}_i)\right)^{\frac{1}{\gamma}} e^{\frac{\sum_i \mu_i \mathbf{h}_i}{\gamma}} \\ &\propto \prod_i \theta_S(\mathbf{h}_0)^{\frac{1}{\gamma}} \left(\theta_E(\mathbf{x}_i|\mathbf{h}_i)e^{\mu_i\mathbf{h}_i}\right)^{\frac{1}{\gamma}} \theta_T(\mathbf{h}_{i+1}|\mathbf{h}_i)^{\frac{1}{\gamma}} \ . \end{aligned}$$

The dual variables-based term can be folded into the emission probabilities, θ_E. Now, the resulting q can be expressed as an HMM by raising θ_S, θ_E, and

[4]The analysis remains the same for a more general formulation with a constant offset vector on the R.H.S. and different matrices for $\mathbf{h}^1$ and $\mathbf{h}^2$.

θ_T to the power $1/\gamma$ and normalizing. For $\gamma = 0$, q can be computed as the most probable output. The required computations in lines 6-9 can be performed using the forward-backward algorithm or the Viterbi algorithm. Note that we can efficiently compute every step because the linear constraints decompose nicely along the probability model.

5 Experiments

Our experiments are designed to explore tuning γ in the UEM framework as a way to obtain gains over EM and hard EM in the constrained and unconstrained cases. We conduct experiments on POS-tagging, word-alignment, and information extraction; we inject constraints in the latter two. In all the cases we use our unified inference step to implement general UEM and the special cases of existing EM algorithms. Since both of our constrained problems involve large scale constrained inference during the E-step, we use UEM$_0$ (with a Lagrange relaxation based E-step) as a proxy for ILP-based CoDL .

As we vary γ over $[0, 1]$, we circumvent much of the debate over EM vs hard EM (Spitkovsky et al., 2010) by exploring the space of EM algorithms in a "continuous" way. Furthermore, we also study the relation between quality of model initialization and the value of γ in the case of POS tagging. This is inspired by a general "research wisdom" that hard EM is a better choice than EM with a good initialization point whereas the opposite is true with an "uninformed" initialization.

Unsupervised POS Tagging We conduct experiments on unsupervised POS learning experiment with the tagging dictionary assumption. We use a standard subset of Penn Treebank containing 24,115 tokens (Ravi and Knight, 2009) with the tagging dictionary derived from the entire Penn Treebank. We run UEM with a first order (bigram) HMM model[5]. We consider initialization points of varying quality and observe the performance for $\gamma \in [0, 1]$.

Different initialization points are constructed as follows. The "posterior uniform" initialization is created by spreading the probability uniformly over all possible tags for each token. Our EM model on this dataset obtains 84.9% accuracy on all tokens and 72.3% accuracy on ambiguous tokens, which is competitive with results reported in (Ravi and Knight, 2009). To construct better initialization points, we train a supervised HMM tagger on hold-out labeled data. The quality of the initialization points is varied by varying the size of the labeled data over $\{5, 10, 20, 40, 80\}$. Those initialization points are then fed into different UEM algorithms.

[5](Ravi and Knight, 2009) showed that a first order HMM model performs much better than a second order HMM model on unsupervised POS tagging

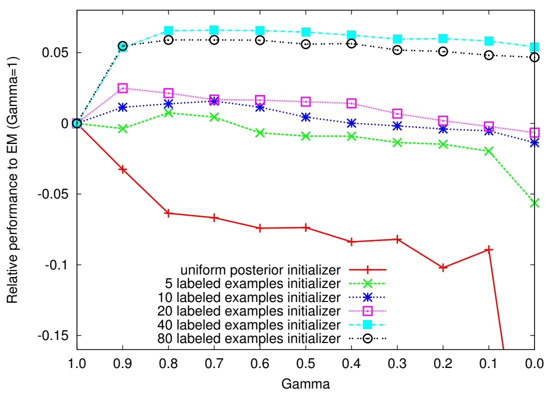

Figure 1: POS Experiments showing the relation between initial model parameters and γ. We report the relative performance compared to EM (see Eq. (16)). The posterior uniform initialization does not use any labeled examples. As the no. of labeled examples used to create the initial HMM model increases, the quality of the initial model improves. The results show that the value of the best γ is sensitive to the initialization point and EM ($\gamma = 1$) and hard EM ($\gamma = 0$) are often not the best choice.

Results For a particular γ, we report the performance of UEM$_\gamma$ w.r.t. EM ($\gamma = 1.0$) as given by

$$rel(\gamma) = \frac{Acc(\text{UEM}_\gamma) - Acc(\text{UEM}_{\gamma=1.0})}{Acc(\text{UEM}_{\gamma=1.0})} \qquad (16)$$

where Acc represents the accuracy as evaluated on the ambiguous words of the given data. Note that $rel(\gamma) \gtrless 0$, implies performance better or worse than EM. The results are summarized in Figure 1.

Note that when we use the "posterior uniform" initialization, EM wins by a significant margin. Surprisingly, with the initialization point constructed with merely 5 or 10 examples, EM is not the best algorithm anymore. The best result for most cases is obtained at γ somewhere between 0 (hard EM) and 1 (EM). Furthermore, the results not only indicate that a measure of "hardness" of EM i.e. **the best value**

of γ, is closely related to the quality of the initialization point but also elicit a more fine-grained relationship between initialization and UEM.

This experiment agrees with (Merialdo, 1994), which shows that EM performs poorly in the semi-supervised setting. In (Spitkovsky et al., 2010), the authors show that hard EM (Viterbi EM) works better than standard EM. We extend these results by showing that this issue can be overcome with the UEM framework by picking appropriate γ based on the amount of available labeled data.

Semi-Supervised Entity-Relation Extraction We conduct semi-supervised learning (SSL) experiments on entity and relation type prediction assuming that we are given mention boundaries. We borrow the data and the setting from (Roth and Yih, 2004). The dataset has 1437 sentences; four entity types: PER, ORG, LOC, OTHERS and; five relation types LIVE IN, KILL, ORG BASED IN, WORKS FOR, LOCATED IN. We consider relations between all within-sentence pairs of entities. We add a relation type NONE indicating no relation exists between a given pair of entities.

We train two log linear models for entity type and relation type prediction, respectively via discriminative UEM. We work in a discriminative setting in order to use several informative features which we borrow from (Roth and Small, 2009). Using these features, we obtain 56% average F1 for relations and 88% average F1 for entities in a fully supervised setting with an 80-20 split which is competitive with the reported results on this data (Roth and Yih, 2004; Roth and Small, 2009). For our SSL experiments, we use 20% of data for testing, a small amount, κ%, as labeled training data (we vary κ), and the remaining as unlabeled training data. We initialize with a classifier trained on the given labeled data.

We use the following constraints on the posterior. 1) Type constraints: For two entities e_1 and e_2, the relation type $\rho(e_1, e_2)$ between them dictates a particular entity type (or in general, a set of entity types) for both e_1 and e_2. These type constraints can be expressed as simple logical rules which can be converted into linear constraints. E.g. if the pair (e_1, e_2) has relation type LOCATED IN then e_2 must have entity type LOC. This yields a logical rule which is converted into a linear constraint as

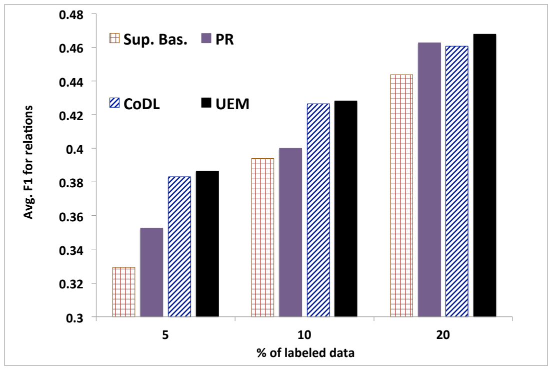

Figure 2: Average F1 for relation prediction for varying sizes of labeled data comparing the supervised baseline, PR, CoDL, and UEM. UEM is statistically significantly better than supervised baseline and PR in all the cases.

$$\begin{aligned}(\rho(e_1, e_2) == \textsc{Located in}) \quad &\rightarrow \quad (e_2 == \textsc{Loc})\\ \Rightarrow q\left(\textsc{Located in}; e_1, e_2\right) \quad &\leq \quad q\left(\textsc{Loc}; e_2\right) \quad .\end{aligned}$$

Refer to (Roth and Yih, 2004) for more statistics on this data and a list of all the type constraints used. 2) Expected count constraints: Since most entity pairs are not covered by the given relation types, the presence of a large number of NONE relations can overwhelm SSL. To guide learning in the right direction, we use corpus-wide expected count constraints for each non-NONE relation type. These constraints are very similar to the *label regularization* technique mentioned in (Mann and McCallum, 2010). Let D_r be the set of entity pairs as candidate relations in the entire corpus. For each non-NONE relation type ρ, we impose the constraints

$$L_\rho \leq \sum_{(e_1,e_2)\in D_r} q(\rho; e_1, e_2) \leq U_\rho \ ,$$

where L_ρ and U_ρ are lower and upper bound on the expected number of ρ relations in the entire corpus. Assuming that the labeled and the unlabeled data are drawn from the same distribution, we obtain these bounds using the fractional counts of ρ over the labeled data and then perturbing it by +/- 20%.

Results We use Alg. 2 for solving the constrained E-step. We report results averaged over 10 random splits of the data and measure statistical significance using paired t-test with $p = 0.05$. The results for relation prediction are shown in Fig. 2. For each trial, we split the labeled data into half to tune the value of γ. For $\kappa = 5\%$, 10%, and 20%, the average value of gamma is 0.52, 0.6, and 0.57, respectively; the median values are 0.5, 0.6, and 0.5, respectively. For relation extraction, UEM is always statistically significantly better than the baseline and PR. The difference between UEM and CoDL is small which is not very surprising because hard EM approaches like CoDL are known to work very well for discriminative SSL. We omit the graph for entity prediction because EM-based approaches do not outperform the supervised baseline there. However, notably, for entities, for $\kappa = 10\%$, UEM outperforms CoDL and PR and for 20%, the supervised baseline outperforms PR statistically significantly.

Word Alignment Statistical word alignment is a well known structured output application of unsupervised learning and is a key step towards machine translation from a source language S to a target language T. We experiment with two language-pairs: English-French and English-Spanish. We use Hansards corpus for French-English translation (Och and Ney, 2000) and Europarl corpus (Koehn, 2002) for Spanish-English translation with EPPS (Lambert et al., 2005) annotation.

We use an HMM-based model for word-alignment (Vogel et al., 1996) and add agreement constraints (Liang et al., 2008; Ganchev et al., 2008) to constrain alignment probabilities in one direction (P_{θ_1}: from S to T) to agree with the alignment probabilities in the other direction (P_{θ_2}: from T to S.) We use a small development set of size 50 to tune the model. Note that the amount of labeled data we use is much smaller than the supervised approaches reported in (Taskar et al., 2005; Moore et al., 2006) and unsupervised approaches mentioned in (Liang et al., 2008; Ganchev et al., 2008) and hence our results are not directly comparable. For the E-step, we use Alg. 3 with R=5 and pick γ from $\{0.0, 0.1, \ldots, 1.0\}$, tuning it over the development set.

During testing, instead of running HMM models for each direction separately, we obtain posterior probabilities by performing agreement constraints-based inference as in Alg. 3. This results in a posterior probability distribution over all possible alignments. To obtain final alignments, following (Ganchev et al., 2008) we use *minimum Bayes risk* decoding: we align all word pairs with posterior marginal alignment probability above a certain threshold, tuned over the development set.

Size	EM	PR	CoDL	UEM	EM	PR	CoDL	UEM
	En-Fr				Fr-En			
10k	23.54	10.63	14.76	**9.10**	19.63	10.71	14.68	**9.21**
50k	18.02	8.30	10.08	**7.34**	16.17	8.40	10.09	**7.40**
100k	16.31	8.16	9.17	**7.05**	15.03	8.09	8.93	**6.87**
	En-Es				Es-En			
10k	33.92	22.24	28.19	**20.80**	31.94	22.00	28.13	**20.83**
50k	25.31	19.84	22.99	**18.93**	24.46	20.08	23.01	**18.95**
100k	24.48	19.49	21.62	**18.75**	23.78	19.70	21.60	**18.64**

Table 2: AER (Alignment Error Rate) comparisons for French-English (above) and Spanish-English (below) alignment for various data sizes. For French-English setting, tuned γ for all data-sizes is either 0.5 or 0.6. For Spanish-English, tuned γ for all data-sizes is 0.7.

Results We compare UEM with EM, PR, and CoDL on the basis of Alignment Error Rate (AER) for different sizes of unlabeled data (See Tab. 2.) See (Och and Ney, 2003) for the definition of AER. UEM consistently outperforms EM, PR, and CoDL with a wide margin.

6 Conclusion

We proposed a continuum of EM algorithms parameterized by a single parameter. Our framework naturally incorporates constraints on output variables and generalizes existing constrained and unconstrained EM algorithms like standard and hard EM, PR, and CoDL. We provided an efficient Lagrange relaxation algorithm for inference with constraints in the E-step and empirically showed how important it is to choose the right EM version. Our technique is amenable to be combined with many existing variations of EM (Berg-Kirkpatrick et al., 2010). We leave this as future work.

Acknowledgments: We thank João Graça for providing the code and data for alignment with agreement. This research is sponsored by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-018, and an ONR Award on Guiding Learning and Decision Making in the Presence of Multiple Forms of Information. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the funding agencies.

References

K. Bellare, G. Druck, and A. McCallum. 2009. Alternating projections for learning with expectation constraints. In *UAI*.

T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. 2010. Painless unsupervised learning with features. In *ACL*, HLT '10.

D. P. Bertsekas. 1999. *Nonlinear Programming*. Athena Scientific, 2nd edition.

P. Brown, S. D. Pietra, V. D. Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*.

M. Chang, L. Ratinov, and D. Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *ACL*.

J. Clarke and M. Lapata. 2006. Constraint-based sentence compression: An integer programming approach. In *ACL*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*.

K. Ganchev, J. Graca, and B. Taskar. 2008. Better alignments = better translations. In *ACL*.

K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*.

T. Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *MlJ*.

M. Kearns, Y. Mansour, and A. Y. Ng. 1997. An information-theoretic analysis of hard and soft assignment methods for clustering. In *ICML*.

D. Klein and C. D. Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *ACL*.

P. Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation.

T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *EMNLP*.

P. Lambert, A. De Gispert, R. Banchs, and J. Marino. 2005. Guidelines for word alignment evaluation and manual alignment. *Language Resources and Evaluation*.

P. Liang, D. Klein, and M. I. Jordan. 2008. Agreement-based learning. In *NIPS*.

G. S. Mann and A. McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 11.

A. Martins, N. A. Smith, and E. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *ACL*.

A. K. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *ICML*.

B. Merialdo. 1994. Tagging text with a probabilistic model. *Computational Linguistics*.

R. C. Moore, W. Yih, and A. Bode. 2006. Improved discriminative bilingual word alignment. In *ACL*.

R. M. Neal and G. E. Hinton. 1998. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*.

K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *ACL*.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *CL*, 29.

S. Ravi and K. Knight. 2009. Minimized models for unsupervised part-of-speech tagging. *ACL*, 1(August).

K. Rose. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *IEEE*, pages 2210–2239.

D. Roth and K. Small. 2009. Interactive feature space construction using semantic information. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

D. Roth and W. Yih. 2004. A linear programming formulation for global inference in natural language tasks. In H. T. Ng and E. Riloff, editors, *CoNLL*.

D. Roth and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*.

A. M. Rush and M. Collins. 2011. Exact decoding of syntactic translation models through lagrangian relaxation. In *ACL*.

A. M. Rush, D. Sontag, M. Collins, and T. Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *EMNLP*.

A. Schrijver. 1986. *Theory of linear and integer programming*. John Wiley & Sons, Inc.

N. A. Smith and J. Eisner. 2004. Annealing techniques for unsupervised statistical language learning. In *ACL*.

V. I. Spitkovsky, H. Alshawi, D. Jurafsky, and C. D. Manning. 2010. Viterbi training improves unsupervised dependency parsing. In *CoNLL*.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *HLT-EMNLP*.

N. Ueda and R. Nakano. 1998. Deterministic annealing em algorithm. *Neural Network*.

S. Vogel, H. Ney, and C. Tillmann. 1996. Hmm-based word alignment in statistical translation. In *COLING*.