Learning Tractable Word Alignment Models with Non-Markovian Constraints

João V. Graça
L²F INESC-ID
Lisboa, Portugal

Kuzman Ganchev
Computer & Information Science
University of Pennsylvania

Ben Taskar
Computer & Information Science
University of Pennsylvania

Word-level alignment of bilingual text is a critical resource for a growing variety of tasks. Word alignment models present a fundamental trade-off between richness of captured constraints and correlations versus efficiency and tractability of inference (Brown et al. 1994; Och and Ney 2003). In this article, we present the Posterior Regularization framework for incorporating prior knowledge constraints during learning without making inference intractable. We focus on the simple and tractable hidden Markov model (Vogel, Ney, and Tillmann 1996), and show an efficient learning algorithm based on constrained bound optimization that incorporates desired non-Markovian constraints such as approximate bijectivity and symmetry. Models estimated with these constraints produce in a significant boost in performance as measured by both precision and recall of manually annotated alignments for six language pairs. We also experiment with the proposed framework on two different tasks where word alignments are required, syntax transfer and phrase-based machine translation, and show promising improvements over standard methods.

1. Introduction

The seminal work of Brown et al. (1994) introduced a series of probabilistic models (IBM models 1-5) for statistical machine translation and the concept of “word-by-word” alignment, the correspondence mapping between words in source and target languages. Although no longer competitive as end-to-end translation models, the IBM models, as well the hidden Markov model (HMM) of Vogel, Ney, and Tillmann (1996), are still widely used for word alignment. Word alignments are used primarily for extracting minimal translation units for machine translation, for example, phrases in phrase-based translation systems (Koehn, Och, and Marcu 2003) and rules in syntax-based machine translation (Galley et al. 2004; Chiang et al. 2005) as well as for MT system combination (Matusov, Ueffing, and Ney 2006). But their importance has grown far beyond machine translation: for instance, transferring annotations between languages by projecting POS taggers, NP chunkers and parsers through word alignment (Yarowsky and Ngai 2001; Rogati, McCarley, and Yang 2003; Hwa et al. 2005; Ganchev, Gillenwater, and Taskar 2009); discovery of paraphrases (Bannard and Callison-Burch 2005; Callison-Burch
2007, 2008); and joint unsupervised POS and parser induction across languages (Snyder and Barzilay 2008; Snyder et al. 2009).

IBM models 1 and 2 and the HMM are simple and tractable probabilistic models, where each target word is generated by a single source word at a time. IBM models 3, 4, and 5 attempt to capture fertility (the tendency of each source word to generate several target words), resulting in probabilistically deficient, intractable models that require local heuristic search and are difficult to implement and extend. Many researchers use the GIZA++ software package (Och and Ney 2003) as a black box, selecting model 4 as a compromise between alignment quality and efficiency. All of the models are asymmetric (switching target and source languages produces drastically different results) and do not enforce approximate bijectivity (the majority of words translating as a single words). In fact although there are systematic situations where one cannot hope to obtain one-to-one alignments, we observe that over 6 different European language pairs the majority of alignments are in fact one-to-one (86%-98%). This leads to the common practice of post-processing heuristics for intersecting directional alignments to produce nearly bijective and symmetric results (Koehn, Och, and Marcu 2003).

In this paper we focus on the HMM word alignment model (Vogel, Ney, and Tillmann 1996), and propose a novel unsupervised learning framework that significantly boosts its performance. The new training framework, called Posterior Regularization (Graça, Ganchev, and Taskar 2008) incorporates prior knowledge in the form of constraints on the posterior distribution over the alignments. This allows the base model to remain tractable, while at the same time approximately satisfying non-Markovian constraints during learning and inference. Two such constraints are described: (i) bijectivity: “one word should not translate to many words”; and (ii) symmetry: “directional alignments should agree”. Both of these constraints significantly improve the performance of the baseline model both in precision and recall, with symmetry constraint generally producing more accurate alignments. In Section 3 we present the Posterior Regularization (PR) framework and how to encode such constraints in an efficient manner which only requires repeated inference in the original model to enforce the constraints.

In Section 4 we compare the alignments obtained by the new training procedure against manually annotated word alignments, and show that constraints over posteriors consistently and significantly outperforms the regular HMM model. Moreover this training procedure in fact outperforms the more complex IBM M4 model 9 out 12 times. We examine the influence of constraints on the resulting posterior distributions and find that they are especially effective for increasing alignment accuracy for rare words.

In Section 5, we evaluate the new framework on two different tasks that depend on word alignments: dependency grammar transfer and phrase-based translation. We consider the task of dependency parsing transfer using the approach proposed in Ganchev, Gillenwater, and Taskar (2009) and show that using the new alignments substantially increases the accuracy of transfer between languages. Subsection 5.2 shows that better alignments also lead to better MT results on an end-to-end phrased-based MT system. As previously shown by Ganchev, Graça, and Taskar (2008) using the new alignments produce consistent improvements over 3 different language pairs, even against IBM M4.

2. Background

Word alignment for a parallel sentence pair represents the correspondence between words in a source language and their translations in a target language (Brown et al. 1994). We will denote each target sentence as \( x = (x_1, ..., x_i, ..., x_I) \) and each source sentence as \( y = (y_1, ..., y_j, ..., y_J) \). A direct correspondence between words in two sentences is not always
Table 1
Test corpora statistics: The last three columns characterize sure alignments: percentage of words that aligned to a single word as well as percentage of words that have n-1 and 1-n alignments, where n ≥ 1.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Sentence Pairs</th>
<th>Ave Length</th>
<th>Max Length</th>
<th>% Sure</th>
<th>% 1-1</th>
<th>% 1-n</th>
<th>% n-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>En/Fr</td>
<td>447</td>
<td>16/17</td>
<td>30/30</td>
<td>21%</td>
<td>96%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>En/Es</td>
<td>400</td>
<td>29/31</td>
<td>90/99</td>
<td>67%</td>
<td>96%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>En/Pt</td>
<td>60</td>
<td>11/11</td>
<td>20/20</td>
<td>54%</td>
<td>91%</td>
<td>94%</td>
<td>96%</td>
</tr>
<tr>
<td>Pt/Es</td>
<td>60</td>
<td>11/11</td>
<td>20/20</td>
<td>69%</td>
<td>92%</td>
<td>96%</td>
<td>93%</td>
</tr>
<tr>
<td>Pt/Fr</td>
<td>60</td>
<td>12/11</td>
<td>20/20</td>
<td>77%</td>
<td>91%</td>
<td>96%</td>
<td>91%</td>
</tr>
<tr>
<td>Es/Fr</td>
<td>60</td>
<td>11/12</td>
<td>20/20</td>
<td>79%</td>
<td>87%</td>
<td>94%</td>
<td>92%</td>
</tr>
</tbody>
</table>

Manual annotated word alignments are required for evaluating the predictive quality of a given alignment model. Due to the inherent ambiguity in word alignments, (e.g. multi-word expressions, annotators dissagreement) it is common to distinguish two kinds of alignments (Och and Ney 2003): sure alignments, for unambiguous alignments, and possible alignments, for ambiguous translations.

We use six manually annotated corpora whose characteristics are summarized in Table 1. The corpora are: the Hansard corpus (Och and Ney 2000) of English/French Canadian Parliamentary proceedings (En-Fr), the English/Spanish portion of the Europarl corpus (Koehn 2002) where the annotation is from EPSS (Lambert et al. 2005) (En-Es) with their usual test and development set split. We also used the English/Portuguese (En-Pt), Portuguese/Spanish (Pt-Es), Portuguese/French (Pt-Fr) and Spanish/French (Es-Fr) portions of the Europarl corpus using annotations described by Graça et al. (2008), where we split the gold alignments into a dev/test set in a ratio of 40%/60%.

Table 1 shows different challenges that each corpus presents. For example, En-Es has longer sentences, which raise the ambiguity level of the alignment task, and smaller percentage of one-to-one alignments. Coping with non-bijective alignments is difficult because of the added uncertainty about word fertility. Another characteristic affecting difficulty is the percentage of sure alignment points out of all alignment points, since in word alignment evaluation only sure points are considered for recall. The range of the percentage of sure alignments varies between 79% for Es-Fr to 21% for En-Fr. Although a direct correspondence between words is often not possible the great majority of alignments are in fact one-to-one (86% - 98%). This characteristic will be explored by the constraints described in this paper.

2.1 HMM word alignment model

In this article we focus on unsupervised training of the hidden Markov model (HMM) for word alignment proposed by Vogel, Ney, and Tillmann (1996). This model generalizes IBM models 1 and 2 (Brown et al. 1994), by introducing a first-order Markov dependence between consecutive target word alignments. The model is an (input-output) HMM with \( I \) positions whose hidden states \( z = (z_1, \ldots, z_I) \) correspond to source word positions (or null to represent
an unaligned target word, \( z_i \in \text{null} \cup \{1, \ldots, J\} \). Each observation corresponds to a word in the target language \( x_i \). Note that this model is directional, each source word (hidden state) can be aligned to at most one target word, leading to \( n-1 \) alignments. From Table 1 one can see the percentage of \( n-1 \) alignments of each corpus, which is an upper bound on the accuracy of directional models.

The probability of an alignment \( z \) and target sentence \( x \) given a source sentence \( y \) can be expressed as:

\[
p_\theta(x, z \mid y) = \prod_{i=1}^{I} p_d(z_i \mid z_{i-1}) p_t(x_i \mid y_{z_i}),
\]

(1)

where \( p_t(x_i \mid y_{z_i}) \) is the probability of a target word at position \( i \) being a translation of the source word at position \( z_i \) (translation probability), and \( p_d(z_i \mid z_{i-1}) \) is the probability of translating a word at position \( z_i \), given that the previous translated word was at position \( z_{i-1} \) (distortion probability). We refer to translation parameters \( p_t \) and distortions parameters \( p_d \) jointly as \( \theta \).

There are several important standard details of the parametrization we follow (Vogel, Ney, and Tillmann 1996): The distortion probability \( p_d(z_i \mid z_{i-1}) \) depends only on the distance \( |z_i - z_{i-1}| \) between the source positions the states represent. Only distances in the range \( \pm 5 \) where explicitly modeled, and larger distances were assigned uniform probabilities. There are also specialized parameters for the first position distortion probability, \( p_d(z_1 \mid z_0) \). To incorporate \( \text{null} \)-alignments, we add a translation probability given \( \text{null} \): \( p_t(x_i \mid y_{\text{null}}) \). Following standard practice of maintaining low distortion across \( \text{null} \)-alignments, we extend the state-space to keep track the previous aligned word when the current word is \( \text{null} \)-aligned and condition on that state for the distortion on the next aligned word.

On the positive side, the model is simple and complexity of inference is \( O(I \times J^2) \). However there are several problems with the model that arise from its directionality.

- **Non-bijective**: Multiple target words can be aligned to a single source word. For instance, the number of one-to-one alignments produced by this model on all available data is 78% instead of the 98% for the Hansards corpus, and for En-Pt corpus its 84% instead of 90%. 4

- **Asymmetric**: Directional models often produce very different alignments, as each model is trained separately and only enforces constraints and correlations between consecutive positions on one side. For instance, if we measure the symmetry between the alignments produced by both directions as the ratio between the number of points in their intersection divided by their union, we get 48% in both the Hansards and En-Pt corpus. 4

### 2.2 Training

Standard model training seeks model parameters \( \theta \) that minimize the negative log-likelihood of the parallel corpus:

\[
\mathcal{L}(\theta) = \mathbb{E}[-\log p_\theta(x \mid y)] = \mathbb{E}[-\log \sum_z p_\theta(x, z \mid y)],
\]

(2)

4 See description in Section 4.
where \( \hat{E}[f(x, y)] = \frac{1}{N} \sum_{n=1}^{N} f(x^n, y^n) \) denotes the empirical average of a function \( f(x^n, y^n) \) over the \( N \) pairs of sentences \( \{(x^1, y^1), \ldots, (x^N, y^N)\} \) in the training corpus. Because of the latent alignment variables \( z \), the negative log-likelihood function for the HMM model is not convex, and the model is fit using Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977). EM minimizes \( \mathcal{L}(\theta) \) via block-coordinate descent on an upper bound \( F(q, \theta) \) using an auxiliary distribution over the latent variables \( q(z | x, y) \) (Neal and Hinton 1998):

\[
\text{EM Upper Bound:} \quad \mathcal{L}(\theta) \leq F(q, \theta) = \hat{E}\left[ \sum_z q(z | x, y) \log \frac{q(z | x, y)}{p(y(x, z | y)} \right].
\]

To simplify notation, we will drop the dependence on \( y \) and will write \( p(y(x, z | y) \) and \( q(z | x, y) \) as \( p(y(x, z) \) and \( q(z | x, y) \). The alternating \( E \) and \( M \) steps at iteration \( t + 1 \) are given by:

\[
E: \quad q^{t+1}(z | x) = \arg\min_{q(z|x)} F(q, \theta^t) = \arg\min_{q(z|x)} \text{KL}(q(z | x) || p(y(x, z | x))) = p(y(x, z | x));
\]

\[
M: \quad \theta^{t+1} = \arg\min_{\theta} F(q^{t+1}, \theta) = \arg\max_{\theta} \hat{E}\left[ \sum_z q^{t+1}(z | x) \log p(y(x, z) \right];
\]

where \( \text{KL}(q | p) = \mathbb{E}_q[\log \frac{q}{p}] \) is the Kullback-Leibler divergence. The EM algorithm is guaranteed to converge to a local minimum of \( \mathcal{L}(\theta) \) under mild conditions (Neal and Hinton 1998). The \( E \) step computes the posteriors \( q^{t+1}(z | x) \) over the latent variables (alignments) given the observed variables (sentence pair) and current parameters \( \theta^t \), which is accomplished by the forward-backward algorithm for HMMs (Rabiner 1989). The \( M \) step uses \( q^{t+1} \) to “softly fill in” the values of alignments \( z \) and estimate parameters \( \theta^{t+1} \). This step is particularly easy for HMMs, where \( \theta^{t+1} \) simply involves normalizing (expected) counts. This modular split into two intuitive and straightforward steps accounts for the vast popularity of EM.

The top row of Figure 1 shows an example of the posterior distribution for the alignment between an English and French sentence. The left figure shows the alignment in the English to French direction where the rows are source words and columns are target words, while the right figure shows the alignment posteriors on the opposite direction. The first observation is that the posteriors are concentrated around particular source words (rare words occurring less than 5 times in the corpus) in both directions, instead of being spread across different words. This is a well known problem when training using EM called “garbage collector effect” (Brown et al. 1993). A rare word in the source language aligns to many words in the target language that we would ideally like to see unaligned, or aligned to other words in the sentence. The reason this happens is because the generative model has to distribute translation probability for each source word among different candidate target words. If one translation is much more common than another, but the rare translation is used in the sentence, the model might have a very low translation probability for the correct alignment. On the other hand, since the rare source word occurs only in a few sentences it needs to spread its probability mass over fewer competing translations. In this case, choosing to align the rare word to all of these words leads to higher likelihood than correctly aligning them or aligning them to the special null word, since it increases the likelihood of this sentence without lowering the likelihood of many other sentences.

### 2.3 Decoding

The alignments are normally predicted using the Viterbi algorithm (which selects the single most probable path through the HMM’s lattice). Another possibility that often works better (Liang, Taskar, and Klein 2006; Graça, Ganchev, and Taskar 2008) is to include the alignment \( i \leftarrow j \)
instead of causing an internal schism.

Figure 1: Posterior distributions for different models for English to French sentence. **Left:** EN—FR model. **Right:** FR—EN model. **Top:** Regular HMM posteriors. **Middle:** After applying bijective constraint. **Bottom:** After applying symmetric constraint. **Sure** alignments are squares with borders; **Possible** alignments are squares without borders. Circle size indicates probability value. Circle color in the middle and bottom rows indicates differences in posterior from the top row. Green - higher probability, red - lower probability.

if the posterior probability that word $i$ aligns to word $j$ is above some threshold. This allows the accumulation of probability from several low-scoring alignments that agree on one point. This accumulated probability (circles in Figure 1) is easily extracted from the model posteriors. Note that this could potentially result in an alignment having zero probability under the model, since $n - m$ alignments can be produced. Posterior decoding has several advantages over Viterbi decoding. First by picking a specific threshold one can trade off precision and recall of the predicted word alignments. Posterior decoding also compensates somewhat for poor estimation of the null word probabilities.

3. **Posterior Regularization**

Word alignment models in general and the HMM in particular are very gross over-simplifications of the translation process and the optimal likelihood parameters learned might not correspond to the optimal alignments. One solution to this problem is to add more complexity to the model to better reflect the translation process. This is the approach taken by IBM models 4+ (Brown et al. 1994) and (Och and Ney 2003), and more recently by the LEAF model (Fraser and Marcu 2007). Unfortunately, these changes make the models probabilistically deficient and intractable, requiring approximations and heuristic learning and inference prone to search errors.

Instead, we use an estimation framework called Posterior Regularization (Graça, Ganchev, and Taskar 2008) that incorporates side-information into unsupervised estimation in the form of linear constraints on posterior expectations. This allows tractable inference even when the constraints would be intractable to encode directly in the model. The following subsections present the Posterior Regularization framework, followed by a description on how to encode two...
pieces of prior knowledge that are hard to encode directly. The first is a bijectivity of alignments – one word should not translate to many words in the other language. The second is a symmetry of alignments – if an English word translates to some French word in a given sentence then the French word should translate as the English word.

3.1 Posterior Regularization Framework

The goal of posterior regularization (PR) framework is to restrict the space of the model posteriors in generative unsupervised learning as a way to guide the model towards desired behavior. Posterior information in PR is specified with sets $Q_x$ of allowed distributions over the hidden variables $z$ defined by linear inequality constraints on feature expectations, with violations bounded by $\epsilon \geq 0$:

\begin{equation}
Q_x = \{ q(z | x) : E_q[f(x, z)] - b_x \leq \xi; \ ||\xi||^2 \leq \epsilon \}.
\end{equation}

$Q_x$ denotes the set of valid distributions where some feature expectations are bounded by $b_x$ and $\epsilon \geq 0$ is an allowable violation slack. Setting $\epsilon = 0$ enforces inequality constraints strictly. In order to introduce equality constraints, we use two inequality constraints with opposite signs. We assume that $Q_x$ is non-empty for each example $x$. Note that $Q_x$, $f(x, z)$ and $b_x$ also depend on $y$, the corresponding source sentence, but we suppress the dependence for brevity. The assumption of linearity of the constraints is computationally important, as will be clear from the resulting algorithm. For now, we do not make any assumptions about the features $f(x, z)$, but we will assume they factor in Eq. 11 for computational efficiency.

In PR, the log-likelihood of a model is penalized with the KL-divergence between the desired distribution space $Q_x$ and the model posteriors, $\text{KL}(Q_x \parallel p_\theta(z | x)) = \min_{q \in Q_x} \text{KL}(q(z) \parallel p_\theta(z | x))$. The regularized objective is:

\begin{equation}
\text{Posterior Regularized Likelihood}:
L(\theta) + \hat{E} [\text{KL}(Q_x \parallel p_\theta(z | x))].
\end{equation}

The objective trades off likelihood and distance to the desired posterior subspace (modulo getting stuck in local maxima) and provides an effective method of controlling the posteriors. The key advantage of using regularization on posteriors is that the learned model itself remains simple and tractable, while during learning, it is driven to obey the constraints through setting appropriate parameters $\theta$. Another advantage is that the objective above can be optimized using a slightly modified EM scheme we describe below (Graça, Ganchev, and Taskar 2008). It also possible to minimize $L(\theta) + \alpha \hat{E} [\text{KL}(Q_x \parallel p_\theta(z | x))]$, for $\alpha \in [0, 1]$ with a very similar scheme.

3.2 Posterior Regularization via Expectation Maximization

Recall that in standard EM E-step, $q(z \parallel x)$ is the posterior over hidden variables given current $\theta$ (Eq. 4). In EM with PR, in order to converge to the local minimum of Eq.7, it suffices to modify the E-step so that $q$ is a projection of the posteriors onto the constraint set $Q_x$ for each example $x$ (see Appendix A for derivation):

\begin{equation}
E' : \arg \min_{q, \xi} \text{KL}(q(z|x) \parallel p_\theta(z|x)) \quad \text{s.t.} \quad E_q[f(x, z)] - b_x \leq \xi; \ ||\xi||^2 \leq \epsilon.
\end{equation}

The new posteriors $q(z|x)$ are used to compute sufficient statistics for this instance and hence to update the model’s parameters in the M-step, which remains unchanged, as in Eq. 5. This scheme is illustrated in Figure 2.
The optimization problem in Eq. 8 can be solved efficiently in dual form (see Appendix B for derivation):

\[
\text{Dual E}': \quad \arg \max_{\lambda \geq 0} -b^\top \lambda - \log Z(\lambda) - \epsilon ||\lambda||_2 ,
\]

(9)

where \( Z(\lambda) = \sum_z p_\theta(z|x) \exp(-\lambda \cdot f(x,z)) \) and the primal solution is \( q(z|x) = p_\theta(z|x) \exp\{-\lambda^\top f(x,z)\}/Z(\lambda) \), where is the normalization constant. There is one dual variable per expectation constraint, and the dual gradient at \( \lambda \neq 0 \) is \( \epsilon \frac{\lambda}{||\lambda||_2} + b - E_q[f(x,z)] \).

When \( \lambda = 0 \), we use sub-gradient optimization and drop the regularization term \( \epsilon \frac{\lambda}{||\lambda||_2} \), which in practice only happens when starting the optimization. When we only have equality constraints, we use the conjugate gradient method for faster convergence, while when there are inequality constraints we use projected gradient descent where each \( \lambda_i \) is truncated to zero when negative. In both methods we use a line search for picking the best step size using strong Wolfe rule (Nocedal and Wright 1999).

Gradient computation involves an expectation under \( q(z|x) \), which can be computed efficiently if the features \( f(x,z) \) factor in the same way as the model \( p_\theta(x,z) \), hence the requirement of linear constraints. The joint distribution over \( z \) represented by a graphical model such as HMM can be written as a product of factors over cliques \( C \):

Factored Posterior: \( p(z|x) = \frac{1}{Z} \prod_{c \in C} \phi(x,z_c) \).

(10)

In an HMM, the cliques \( C \) are simply the nodes \( z_i \) and the edges \( (z_i, z_{i+1}) \) and the factors correspond to the distortion and translation probabilities. We will assume \( f \) is factorized along the same cliques (we will show below how symmetry and bijectivity constraints can be expressed in this way):

Factored Features: \( f(x,z) = \sum_{c \in C} f(x,z_c) \).

(11)

Then \( q(z|x) \) has the same form as \( p_\theta(z|x) \):

\[
q(z|x) = \frac{1}{Z} p(z|x) \exp(-\lambda^\top f(x,z)) = \frac{1}{Z} \prod_{c \in C} \phi(x,z_c) \exp(-\lambda^\top f(x,z_c)) .
\]

(12)

Hence the projection step uses the same inference algorithm (forward-backward for HMMs) to compute the gradient, only modifying the local factors using the current setting of \( \lambda \).
3.3 Bijectivity Constraints

Bijectivity constraints are based on the observation that in most gold alignments, words are aligned as one-to-one. We would like to introduce this trend into the model, but adding this to the model directly breaks the Markov property. In fact, summing over one-to-one or near one-to-one weighted matchings is a classical #P-Complete problem (Valiant 1979). However, introducing alignment degree constraints on expectation in the PR framework is easy and tractable. We simply introduce inequality constraints $E[f(x, z)] \leq 1$ where we have one feature for each source word $j$ that counts how many times it is aligned to a target word in the alignment $z$:

Bijective Features: $f_j(x, z) = \sum_i 1(z_i = j)$.

For example for the sentence in Figure 1 in the top right alignment the posteriors over the source word schism clearly sum to more than 1. The effects of applying these constraints to the posteriors are shown in the second row. Enforcing the one to (at most) one constraint clearly solves the garbage collector effect. Moreover, when distributing the probability mass to the other words, most of the probability mass goes into the right positions (as measured by gold alignments). By enforcing the constraint at training and decoding time the percentage of 1-1 alignments increases from 78% on Hansards English to French to 97.3% (manual alignments have 98.1%), and in the En-Pt corpus, the constraint improves from 84.7% to 95.8% (manual alignments have 90.8%).

3.4 Symmetry Constraints

Word alignment should not depend on translation direction, which is clearly violated by the directional models like the HMM. In fact, each directional model makes different mistakes (see Figure 1). The standard approach is to train two models independently and then intersect their predictions (Och and Ney 2003). However, we show that it is much better to train two directional models concurrently, coupling their posterior distributions over alignments to approximately agree. The idea of training jointly has been explored by Matusov, Zens, and Ney (2004) and Liang, Taskar, and Klein (2006), although their formalization is quite different.

Let the directional models be defined as: $p^>(z)$ (forward) and $p^<(z)$ (backward). We suppress dependence on $x$ and $y$ for brevity. Define $z$ to range over the union of all possible directional alignments $\overline{Z} \cup \overline{Z}$. We define a mixture model $p(z) = \frac{1}{2} p^>(z) + \frac{1}{2} p^<(z)$ where $p^>(z) = 0$ and vice-versa (i.e., the alignment of one directional model has probability zero according to the other model). We then define the following feature for each target-source position pair $i, j$:

Symmetric Features: $f_{ij}(x, z) = \begin{cases} +1 & z \in \overline{Z} \text{ and } \overline{z}_i = j \\ -1 & z \in \overline{Z} \text{ and } \overline{z}_j = i \\ 0 & \text{otherwise} \end{cases}$.

The feature takes the value zero on expectation if a word pair $i, j$ is aligned with equal probability in both directions. So the constraints we want to impose is $E_q[f_{ij}(x, z)] = 0$ (possibly with some small violations). Note that this constraint is only feasible if the posteriors are bijective. Clearly these features are fully factored. To compute expectations of these features under the model $q$ we only need to be able to compute them under each directional HMM, as we show below. To see this, we have by the definition of $q_\lambda$ and $p_\theta$,

$$q_\lambda(z|x) = \frac{\overline{p}^>(z|x) + \overline{p}^<(z|x) \exp\{-\lambda^T f(x, z)\}}{2Z} = \frac{\overline{q}^>(z|x) Z^>_{\overline{x}} + \overline{q}^<(z|x) Z^<_{\overline{x}}}{2Z},$$

(13)
where we have defined:

\[
\overrightarrow{q}(z|x) = \frac{1}{Z_{\overrightarrow{q}}} \overrightarrow{p}(z, x) \exp\{-\lambda^T f(x, z)\}
\]

with

\[
Z_{\overrightarrow{q}} = \sum_z \overrightarrow{p}(z, x) \exp\{-\lambda^T f(x, z)\}
\]

\[
\overleftarrow{q}(z|x) = \frac{1}{Z_{\overleftarrow{q}}} \overleftarrow{p}(z, x) \exp\{-\lambda^T f(x, z)\}
\]

with

\[
Z_{\overleftarrow{q}} = \sum_z \overleftarrow{p}(z, x) \exp\{-\lambda^T f(x, z)\}
\]

All these quantities can be computed separately in each model using forward-backward and furthermore,

\[
Z = \frac{1}{2} \left( Z_{\overrightarrow{q}} + Z_{\overleftarrow{q}} \right).
\]

Figure 1 shows the expected value of each feature as the difference between the posterior circles on both directional models for the same word pair. The last row shows both directional posteriors after imposing the symmetric constraint. Note that the projected posteriors are equal in the two models. Also, one can see that in most cases the probability mass was moved into the right place with the exception of the word pair \textit{internal}/\textit{le}; this is because the word \textit{internal} does not appear on the French side, but the model still has to spread around the probably mass to that word. In this case the model decided to accumulate it into the word \textit{le} instead of moving it to the \textit{null} word. By enforcing this constraint at training and decoding time we can improve the symmetry of the directional alignments (as measured by intersection over union) on the Hansards corpus from 48\% to 89.9\% and on the En-Pt corpus from 48\% to 94.2\%.  

4. Word Alignment Evaluation

We begin with a comparison of word alignment quality evaluated against manual annotated alignments and measured by precision and recall. We use the six parallel corpus with gold annotations described in the beginning of section 2.

4.1 Experimental Setup

The results reported in the sequel were generated in the following manner. We discarded all training data sentences of length over 40 and, following common practice, added the unlabeled development and test data sets to the pool of unlabeled sentences. We initialized an IBM Model 1 translation table with uniform probabilities over word pairs that occur together in same sentence and trained Model 1 for 5 iterations.

All HMM aligners were initialized with the final translation table from Model 1 and uniform distortion probabilities and trained until the area under the precision/recall curve stopped increasing (see Figure 3 for an example of such a curve). In most cases this meant 4 iterations for normal EM training and 2 iterations for posterior regularization. We suspect that the constraints make the space easier to search. Unfortunately this does not result in a decreased running time because projecting the model posteriors is computationally expensive. We perform this projection until convergence or a maximum number of iterations. Here, convergence is measured with respect to the normalized \(l_2\) norm of the gradient (gradient norm divided by number of constrains). We call the maximum allowable value of this \(l_2\) norm a convergence tolerance to distinguish it from the slack described in Section 3. For the bijective constraint we set tolerance to 0.005, slack to 0 and maximum number of iterations to 1000. For symmetric tolerance and slack were 0.001 and the maximum number of iterations was 1000. We chose the maximum number of iterations so that most sentences can converge. Setting tolerance very high ensures fast convergence, but the results become more similar with the baseline HMM. The improved optimization of the projection has a significant impact on alignment quality, and explains why the results we report here are better than previous work that used a more naïve optimization (Ganchev, Graça, and Taskar 2008).
Figure 3

We set tolerance to an aggressively small value in the sense that smaller values of tolerance did not produce a significant increase in alignment quality and were much more computationally expensive.

We also present results for IBM M4 using the standard alignment toolkit, GIZA++ (Och and Ney 2003). Training was done using the default configuration of the MOSES training script\(^5\). This performs 5 iterations of M1, 5 iterations of HMM and 5 iterations of IBM M4. Since IBM M4 is more complicated and can capture more structure, but at the same does not allow the use of posterior decoding, the fair comparison is between the performance of the baseline HMM and the constraint trained HMM. Note also that there might be slight implementation divergences between GIZA++ used to run model 4 and POST-CAT\(^6\) (Graça, Ganchev, and Taskar 2009) used to run all the other models.

4.2 Overall Results

In this subsection we present overall results of alignment quality. For the models with constraints we project the posteriors at decode time. This gives a small but consistent improvement. Figure 3 shows precision/recall curves for the different models on the En-Fr corpus in the English-French direction (left), and on the En-Pt corpus Portuguese-English direction (right). All comparisons are made using posterior decoding since this decoding method always outperforms Viterbi decoding.

We can read off several properties from Figure 3. Firstly, both constraints improve over the baseline HMM in terms of both precision and recall. When compared against IBM Model 4, the constraints based model present better results for the En-Fr corpus (the lines are strictly above the M4 point), and the symmetric training is slightly better than the En-Pt corpus.

In order to make the models easier to compare concisely we implemented a decoding method that chooses a threshold to achieve the recall of Model 4. Figure 4 compares the different models for our six corpora in both directions. We see that the constraint based methods always improve over the regular HMM, and in 9 out of 12 cases, they outperform Model 4.

Figure 5 shows performance as a function of training data size. In particular, we fix recall at the value achieved by Model 4 and measure precision as we vary the amount of training data.

\(^5\) http://www.statmt.org/moses/?n=FactoredTraining.HomePage
\(^6\) http://www.seas.upenn.edu/~strctlrn/CAT/
Figure 4
Word alignment precision when the threshold is chosen to achieve Model 4 recall with a difference of +/- 0.005. The average relative increase in precision (against the HMM model) is 10% for M4, 11% for B-HMM and 14% for S-HMM.

Figure 5

data in number of sentences. For small training corpora adding the constraints provides larger improvements (20-30)% but we still achieve significant gains even with a million parallel sentences (15%). These great improvements for small data are not surprising, but show that our approach is most useful when few data are available, such as a domain with small corpora or a resource-poor language pair.

4.3 Rare vs Common words

One of the main benefits of using the posterior regularization constraints described is an alleviation of the garbage collector effect (?). Figure 6 breaks down performance improvements by common versus rare words. As before, we use posterior decoding, tuning the threshold to match M4 recall. For common words, this tuning maintains recall very close for all models so we do not show this in the figure.

In the left panel of Figure 6, we see that precision of common words follows the pattern we saw for the corpus overall: symmetric and bijective outperform both Model 4 and the baseline HMM, with symmetric slightly better than bijective. The results for common words vary more slowly as we increase the quantity of training data than they did for the full corpus. In the middle
panel of Figure 6 we show the precision for rare words. For the baseline HMM as well as for Model 4, this is very low precisely because of the garbage collector problem: rare words become erroneously aligned to untranslated words, leading to low precision. In fact the constrained models achieve absolute precision improvements of up to 50% over the baseline. By removing these erroneous alignments the translation table becomes cleaner allowing higher recall on the full corpus. In the right panel of Figure 6, we observe a slightly diminished recall for rare words. This slight drop in recall is due to moving the mass corresponding to rare words to null.

4.4 Symmetrization

As discussed earlier, the word alignment models are asymmetric, while most applications require a single alignment for each sentence pair. Typically this is achieved by a symmetrization heuristic that takes two directional alignments and produces a single alignment. For MT the most commonly used heuristic is called “grow diagonal final” (Och and Ney 2003). This starts with the intersection of the sets of aligned points and adds points around the diagonal that are in the union of the two sets of aligned points. The alignment produced has high recall relative to the intersection and only slightly lower recall than the union. In syntax transfer the intersection heuristic is normally used, since one wants to have highly precise links to transfer knowledge between languages. One pitfall of these symmetrization heuristics, is that they can obfuscate the link between the original alignment and the ones used for a specific task, making errors more difficult to analyze. Since they are heuristic it is not clear when they will help and when they will hinder system performance.

Figure 6
Precision and Recall as a function of training data size for En-Fr by common and rare words. Left: Common Precision, Middle: Rare Precision. Right Rare Recall.

Figure 7
Precision/recall curves for the different models after soft union symmetrization. Precision is on the horizontal axis.

Figure 8
Precision/recall curves for symmetry constraints based model. Both directional models curves are very close to the one after performing the symmetrization heuristic.
In this work we followed a more principled approach that uses the knowledge about the posterior distributions of each directional model. We include a point in the final alignment if the average of the posteriors under the two models for that point is above a threshold. This heuristic is called soft union (DeNero and Klein 2007). Figure 7 shows the precision/recall curves after symmetrization for the En-Fr corpus. The posterior regularization trained models still performed better, but the differences get smaller after doing the symmetrization as expected. The explanation for this is that this symmetrization is similar to using our symmetry constrains at decoding time. It is also very similar to the symmetric alignments of (Liang, Taskar, and Klein 2006), except that (Liang, Taskar, and Klein 2006) use soft intersection rather than soft union. Figure 8 shows that after performing the projection step, applying a symmetrization heuristic to the symmetric constrained model does not change performance significantly. This is not surprising because the model posteriors are similar after the projection step. The small deviations are because of incomplete convergence and slack as discussed earlier.

4.5 Analysis

In this section we discuss some scenarios in which the constraints make the alignments better, and some scenarios where they fail. We have already discussed the garbage collector effect and how both models address it. Both of the constraints also bias the model to have at most probability one in any row or column of the posterior matrix, encouraging 1-1 alignments. Obviously whenever alignments are systematically not 1-1, this can lead to errors (for instance the example described at the end of section 2.1).

Figure 9
Posterior distributions for different models for English to French sentence. Left: EN→FR model. Right: FR→EN model. Top: Regular HMM posteriors. Middle: After applying bijective constraint. Bottom: After applying symmetric constraint. Sure alignments are squares with borders; possible alignments are squares without borders. Circle size indicates probability value. Circle color in the middle and bottom rows indicates differences in posterior from the top row. Green - higher probability, red - lower probability.
An example is presented in Figure 9, where we show the posterior distributions for an English/Spanish sentence pair using the same notation as in Figure 1. In the top panel of Figure 9, we see the baseline models, where the English word *met* is incorrectly being aligned to *séance est ouverte*. This makes it impossible to recover the correct alignment *house/séance*. Either constraint corrects this problem. On the other hand by enforcing a 1-1 mapping the correct alignment *met / est ouverte* is lost. Going back to the first row (regular HMM) this alignment is correct in one direction and absent in the other (due to the n-1 model restriction) but we can recover that information using the symmetrization heuristics, since the point is present at least in one direction with high probability mass. This is not the case for the constraint based models that reduce the mass of that alignment in both directions.

There are two possible solutions to alleviate this type of problem, both with their caveats. One solution is to model the fertility of each word in a way similar to IBM Model 4, or more generally to model alignments of multiple words. This can lead to significant computational burden, and is not guaranteed to improve results. It might be the case that a more complicated model requires approximations that destroy its performance gain, or requires larger corpora to estimate its parameters. Another option is to perform some linguistically motivated pre-processing of the language pair. This of course has the disadvantage that it needs to be specific to a language pair, in order to include information such as “English past perfect is written using a single word, so join together French passé composé.” An additional problem with joining words to alleviate inter-language divergences is that it can increase data sparsity.

5. External Word Alignment Evaluation

In this section we evaluate the alignments resulting from using the proposed constraints in two different tasks: Syntax transfer where alignments are used to indicate which labels should match in the different languages; Statistical machine translation where alignments are used to restrict the number of possible minimal translation units.

5.1 Syntax Transfer

Here we compare the different alignment models based on how well they can be used for transfer of linguistic resources across languages. In particular, (Ganchev, Gillenwater, and Taskar 2009) use a word aligned corpus and a parser for a resource rich language (source language) in order to create a parser for a resource poor language (target language). Consider the parse tree of the source language as a set of dependency edges. For each such edge, (Ganchev, Gillenwater, and Taskar 2009) check whether both end points are aligned to words in the target language. If both endpoints are aligned, then the edge is transferred. Obviously, not all edges transferred in this way result in correct parse edges. Parser errors in the source language can result in incorrect parses. Additionally, linguistic phenomena in the two languages might cause correct parses and alignments to result in incorrect transferred edges. For example the verb “bike” in English might be translated to French as “aller à vélo” where the semantic content is translated as the word “vélo” but we cannot expect this French noun to behave similarly to the English verb. In order to address this concern, (Ganchev, Gillenwater, and Taskar 2009) filter the resulting alignments by POS tag.

In order to evaluate the alignments without dependence on a particular parser, we computed the fraction of correctly transferred edges as a function of the average number of edges transferred. In general as trade off precision vs recall of alignments, we can increase the accuracy of the transferred edges by transferring a smaller number of edges. Figure 10 shows our results for transferring from English to Bulgarian (En→Bg) and from English to Spanish (En→Es). The En→Bg results are based on a corpus of movie subtitles, and are consequently shorter.
sentences while the En→Es are based on a corpus of parliamentary proceedings. We use the “two-rules” system from (Ganchev, Gillenwater, and Taskar 2009) for Bulgarian, using sentences of any length that have at least one transferred edge. For Spanish we use the “three-rules” system from (Ganchev, Gillenwater, and Taskar 2009). See (Ganchev, Gillenwater, and Taskar 2009) for details on the corpora and transfer rules. The alignments were produced using the softUnion heuristic. Figure 10 shows the percentage of transferred edges that are correct, as a function of the average number of edges per sentence that were transferred for the two languages (for different values of the posterior threshold). The Bulgarian corpus has shorter sentences on average, resulting in a smaller number of edges transferred per sentence and in a higher accuracy relative to Spanish. We see in Figure 10 that for both domains, the models trained using posterior regularization perform better than the baseline model trained using EM.

5.2 Phrase-based machine translation

In this subsection we attempt to investigate whether the constraint alignments produce improvements in a end to end phrase based machine translation system. In particular we fix a state of the art machine translation system, the open source Moses (Koehn et al. 2007) toolkit7, and measure its performance when we vary the supplied word alignments.

For all experiments the experimental setup is as follows: we lowercase the corpora, and train language models from all available data. The reasoning behind this is that even if bilingual texts might be scarce in some domain, monolingual text should be relatively abundant. For each competing alignment model we train Moses and use MERT optimization to tune its parameters on a development set. Moses is trained starting from a symmetrized alignment set, phrases up to 7 words are extract, and the msd-bidirectional-fe distance based distortion model is used. The competing alignment models are GIZA Model 4, where the directional models are symmetrized using the grow diagonal and final heuristic (M4 GDF), our implementation of the baseline HMM alignment using both the grow diagonal final heuristic (HMM GDF) and the softUnion heuristic (HMM SU) and the HMM trained with both constraints using the softUnion heuristic (B-HMM SU, S-HMM SU). When using the softUnion heuristic we tested three values of the threshold (0.2,0.4,0.6) which try to capture the choice between recall oriented alignments and precision oriented alignments, and pick the best according to the performance in the development set. Table 5.2 summarizes the results for the different corpora and alignments.

---

7 www.statmt.org/moses/
A first observation is that the constrain based models always outperform HMM (both GDF and SU) and outperform M4 GDF in all but one experiment, with a difference ranging from (0.2 to 0.9). However both constrains help in different cases. While the Bijectivity constrains seem to help more for the Hansards corpus and for one direction of the EPPS corpus, in the other cases the symmetry constrains outperform all other models. Regarding the particular choice of symmetrization heuristic we see that the softUnion beats the grow diagonal and final in 3 out of 4 cases. It is also interesting that the baseline HMM with softUnion outperforms the more complex M4 baseline in half the cases. Even if the constrained alignments consistently lead to better BLEU scores this differences are small and not easy to justify. There are several influencing factors, related with the high entropy on the typical MT cascade: There was a huge variance in the number of iterations MERT required to converge ranging from just 2 iterations to 28 iterations, also the best value on the development set did not always coincide with the best value on the test set. Moreover, against common knowledge in MT community which indicates that, bigger phrase tables are better, we note that in 14 out of 18 times the threshold picked was 0.4 (middle size phrase tables) and the other 4 times 0.2 was picked (smaller phrase tables). For instance for the symmetric HMM model on the en-pt corpus the phrase table sizes range from (0.8 GM for 0.2, 1.7G for 0.4 and 2.7G for 0.6). This is probably related with the fact that the number of aligned points with bigger thresholds is small leading to a lot of erroneous phrases constituted mostly by unaligned points, which will lead to a poor estimation of phrase probabilities, as noted also in Lopez and Resnik (2006).

6. Conclusion and Future Work

In this paper we exploited a novel learning framework, Posterior Regularization, for incorporating rich constraints over the posterior distributions over word alignments. We focused on the HMM word alignment model, and showed how we could incorporate non-Markovian constraints like bijectivity and symmetry while at the same time keeping the inference in the model tractable. Using these constraints we showed consistent and significant improvements in 6 different languages pairs even when compared to a more complex model such as IBM M4. Besides solving the known “garbage collector” effect, we show that the posterior distributions obtained better modeled the desired alignments. Given the intuition that both constrains are biasing the models for 1-1 alignments, we also show some systematic mistakes that the constrains introduce and suggest future work on how to solve them.

We experimented with two different tasks that rely on word alignments, syntax transfer and statistical phrase based MT, where the improved alignments lead to an increase in performance. In the case of syntax transfer, we shown that the number of edges of a dependency tree that can be transfer from one language to the other increases due to the decrease of incorrect alignment
points. On phrase based MT, the improvements are harder to explain, but in fact over three
different language pairs we get improvements using the new alignments. We note however that
the current heuristics for phrase extraction and phrase scoring do not take into account the quality
(or probability) of the base alignments, and aim at achieving as many phrases as possible (leading
to huge phrase tables), which obfuscates the relation between word alignments and MT quality.

A drawback with the current framework implementation is the time taken for projection the
posterior distribution into the constraint space. One approach to speeding up the E-step is to use
fully factored variational inference since full HMM inference is expensive and is currently run
multiple times in linesearch. Another speedup is expected by moving from the regular batch EM
towards a online version, due to redundancy in a large parallel corpus.

Beyond the two constraints discussed, many others are worth investigating. To simulate
fertility modeling one could replace the bijectivity constraint by a fertility constraint where the
probability mass of each word had to sum not to one, but to a constant depending of the word
identity or the word class. A distortion constraint could be used to penalize the distance between
words according to a dependency tree rather then the absolute position.

Acknowledgments

J. V. Graça was supported by a fellowship from Fundação para a Ciência e Tecnologia (SFRH/
BD/ 27528/ 2006). K. Ganchev was partially supported by NSF ITR EIA 0205448.

References

1993. But dictionaries are data too. In Proc. HLT.
EMNLP.
Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the
EMNLP-CoNLL.
HLT-NAACL.
constraints. In Proc. ACL.
Proc. NIPS.
Prague Bulletin Of Mathematical Linguistics - Special Issue: Open Source Tools for Machine
Translation, 91:27–37, January.
multi-language word alignment. In Proc LREC.
Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen,
ACL.

Appendix A: Convergence of PR EM

We show that minimizing constrained $F(q, \theta)$ results in minimizing the PR Objective.

**Theorem:** $F(q, \theta)$ and PR

Local minima of $F(q, \theta)$ s.t. $q(z \mid x) \in Q_x$ are local minima of $\mathcal{L} (\theta) + \hat{\mathbf{E}}[\text{KL}(Q_x \parallel p_\theta(z \mid x))]$.

**Proof:** By adding and subtracting $\hat{\mathbf{E}}[\sum_x q(z \mid x) \log p_\theta(z \mid x)]$ from $F(q, \theta)$, we get:

$$F(q, \theta) = \hat{\mathbf{E}} \left[ \sum_x q(z \mid x) \log \frac{q(z \mid x)}{p_\theta(z \mid x)} \right] = -\hat{\mathbf{E}} \left[ \sum_x q(z \mid x) \log \frac{p_\theta(z \mid x)}{p_\theta(z \mid x)} \right] + \hat{\mathbf{E}} \left[ \sum_x q(z \mid x) \log \frac{q(z \mid x)}{p_\theta(z \mid x)} \right]$$

Since the first term does not depend on $q$, the second term is minimized by $q^*(z \mid x) = \arg\min_{q(z \mid x) \in Q_x} \text{KL}(q(z \mid x) \parallel p_\theta(z \mid x))$ at local minima.

**Appendix B: Modified E-step dual derivation**

The modified E-step involves a projection step that minimizes the Kullback-Leibler divergence:

$$\mathbf{E}': \arg\min_q \text{KL}(q(z \mid x) \parallel p_\theta(z \mid x)) \quad \text{s.t.} \quad \mathbf{E}_q[f(x, z)] - b_x \leq \zeta; \quad \|\xi\|_2^2 \leq \epsilon.$$
Assuming the set $Q_x = \{ q(z|x) : E_q[f(x,z)] - b_x \leq \xi; \|\xi\|_2^2 \leq \epsilon \}$ is non-empty, the corresponding Lagrangian is

$$L(q(z|x), \xi, \lambda, \alpha, \gamma) = \max_{\lambda,\alpha,\gamma} \min_{q(z|x),\xi} L(q(z|x), \xi, \lambda, \alpha, \gamma),$$

where

$$L(q(z|x), \xi, \lambda, \alpha, \gamma) = \text{KL}(q(z|x) \parallel p_\theta(z|x)) + \lambda^T (E_q[f(x,z)] - b_x - \xi)$$

$$+ \alpha(||\xi||_2^2 - \epsilon^2) + \gamma(\sum_z q(z|x) - 1)$$

and

$$\frac{\partial L(q(z|x), \xi, \lambda, \alpha, \gamma)}{\partial q(z|x)} = \log(q(z|x)) + 1 - \log(p_\theta(z|x)) + \lambda^T f(x,z) + \gamma = 0$$

$$\Rightarrow q(z|x) = \frac{p_\theta(z|x) e^{-\lambda^T f(x,z)}}{\exp(\gamma)}$$

Plugging $q(z|x)$ and $\xi$ in $L(q(z|x), \xi, \lambda, \alpha, \gamma)$ and taking the derivative with respect to $\gamma$.

$$\frac{\partial L(\lambda, \alpha, \gamma)}{\partial \gamma} = \sum_z \frac{p_\theta(z|x) e^{-\lambda^T f(x,z)}}{e \exp(\gamma)} - 1 = 0 \Rightarrow \gamma = \log\left(\sum_z p_\theta(z|x) e^{-\lambda^T f(x,z)}\right)$$

From where we can simplify $q(z|x) = \frac{p_\theta(z|x) e^{-\lambda^T f(x,z)}}{Z_\lambda}$ where $Z_\lambda = \sum_z p_\theta(z|x) e^{-\lambda^T f(x,z)}$ ensures that $q(z|x)$ is properly normalized. Plugging $\gamma$ into $L(\lambda, \alpha, \gamma)$ and taking the derivative with respect to $\alpha$, we get:

$$L(\lambda, \alpha) = -\log(Z) - b_x^T \lambda - \frac{\lambda^2}{2\alpha} - \frac{\lambda^2}{4\alpha} - \alpha \epsilon^2$$

(B.1)

$$\frac{\partial L(\lambda, \alpha)}{\partial \alpha} = \frac{\lambda^2}{2\alpha^2} - \frac{\lambda^2}{4\alpha^2} - \epsilon^2 = 0 \Rightarrow \alpha = \frac{||\lambda||_2}{2\epsilon}$$

(B.2)

Replacing back into $L(\lambda, \alpha)$ we get the dual objective:

$$\text{Dual } \mathcal{E}': \arg \max_{\lambda \geq 0} -b_x^T \lambda - \log(Z) - ||\lambda||_2 \epsilon$$

(B.3)