



University of Pennsylvania
ScholarlyCommons

Lab Papers (GRASP)

General Robotics, Automation, Sensing and
Perception Laboratory

3-10-2010

Learning Tractable Word Alignment Models with Complex Constraints

João V. Graça
INESC-ID Lisboa

Kuzman Ganchev
University of Pennsylvania, kuzman@seas.upenn.edu

Ben Taskar
University of Pennsylvania, taskar@cis.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/grasp_papers

 Part of the [Engineering Commons](#)

Recommended Citation

João V. Graça, Kuzman Ganchev, and Ben Taskar, "Learning Tractable Word Alignment Models with Complex Constraints", . March 2010.

Suggested Citation:

J. Graça, K. Ganchev and B. Taskar. (2010). "Learning Tractable Word Alignment Models with Complex Constraints." *Computational Linguistics*. Vol. 36(3). p. 481-504.

© 2010 MIT Press

<http://www.mitpressjournals.org/loi/coli>

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/grasp_papers/66

For more information, please contact libraryrepository@pobox.upenn.edu.

Learning Tractable Word Alignment Models with Complex Constraints

Abstract

Word-level alignment of bilingual text is a critical resource for a growing variety of tasks. Probabilistic models for word alignment present a fundamental trade-off between richness of captured constraints and correlations versus efficiency and tractability of inference. In this article, we use the Posterior Regularization framework (Graça, Ganchev, and Taskar 2007) to incorporate complex constraints into probabilistic models during learning without changing the efficiency of the underlying model. We focus on the simple and tractable hidden Markov model, and present an efficient learning algorithm for incorporating approximate bijectivity and symmetry constraints. Models estimated with these constraints produce a significant boost in performance as measured by both precision and recall of manually annotated alignments for six language pairs. We also report experiments on two different tasks where word alignments are required: phrase-based machine translation and syntax transfer, and show promising improvements over standard methods.

Disciplines

Engineering

Comments

Suggested Citation:

J. Graça, K. Ganchev and B. Taskar. (2010). "Learning Tractable Word Alignment Models with Complex Constraints." *Computational Linguistics*. Vol. 36(3). p. 481-504.

© 2010 MIT Press

<http://www.mitpressjournals.org/loi/coli>

Learning Tractable Word Alignment Models with Complex Constraints

João V. Graça*
L²F INESC-ID

Kuzman Ganchev**
University of Pennsylvania

Ben Taskar†
University of Pennsylvania

Word-level alignment of bilingual text is a critical resource for a growing variety of tasks. Probabilistic models for word alignment present a fundamental trade-off between richness of captured constraints and correlations versus efficiency and tractability of inference. In this article, we use the Posterior Regularization framework (Graça, Ganchev, and Taskar 2007) to incorporate complex constraints into probabilistic models during learning without changing the efficiency of the underlying model. We focus on the simple and tractable hidden Markov model, and present an efficient learning algorithm for incorporating approximate bijectivity and symmetry constraints. Models estimated with these constraints produce a significant boost in performance as measured by both precision and recall of manually annotated alignments for six language pairs. We also report experiments on two different tasks where word alignments are required: phrase-based machine translation and syntax transfer, and show promising improvements over standard methods.

1. Introduction

The seminal work of Brown et al. (1993b) introduced a series of probabilistic models (IBM Models 1–5) for statistical machine translation and the concept of “word-by-word” alignment, the correspondence between words in source and target languages. Although no longer competitive as end-to-end translation models, the IBM Models, as well as 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 (MT) (e.g., phrases [Koehn, Och, and Marcu 2003] and rules [Galley et al. 2004; Chiang et al. 2005]) as well as for

* INESC-ID Lisboa, Spoken Language Systems Lab, R. Alves Redol 9, 1000-029 LISBOA, Portugal.
E-mail: joao.graca@l2f.inesc-id.pt.

** University of Pennsylvania, Department of Computer and Information Science, Levine Hall, 3330 Walnut Street, Philadelphia, PA 19104-6309. E-mail: kuzman@cis.upenn.edu.

† University of Pennsylvania, Department of Computer and Information Science, 3330 Walnut Street, Philadelphia, PA 19104-6389. E-mail: taskar@cis.upenn.edu.

Submission received: 1 August 2009; revised submission received: 24 December 2009; accepted for publication: 10 March 2010.

MT system combination (Matusov, Ueffing, and Ney 2006). But their importance has grown far beyond machine translation: for instance, transferring annotations between languages (Yarowsky and Ngai 2001; Hwa et al. 2005; Ganchev, Gillenwater, and Taskar 2009); discovery of paraphrases (Bannard and Callison-Burch 2005); and joint unsupervised POS and parser induction across languages (Snyder and Barzilay 2008).

IBM Models 1 and 2 and the HMM are simple and tractable probabilistic models, which produce the target sentence one target word at a time by choosing a source word and generating its translation. 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 IBM 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 the simpler models (IBM Models 1, 2, and HMM) do not enforce bijectivity (the majority of words translating as a single word). Although there are systematic translation phenomena where one cannot hope to obtain 1-to-1 alignments, we observe that in over 6 different European language pairs the majority of alignments are in fact 1-to-1 (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 article we focus on the HMM word alignment model (Vogel, Ney, and Tillmann 1996), using a novel unsupervised learning framework that significantly boosts its performance. The new training framework, called Posterior Regularization (Graça, Ganchev, and Taskar 2007), incorporates prior knowledge in the form of constraints on the model's posteriors. The constraints are expressed as inequalities on the expected value under the posterior distribution of user-defined features. Although the base model remains unchanged, learning guides the model to satisfy these constraints. We propose two such constraints: (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 model both in precision and recall, with the symmetry constraint generally producing more accurate alignments. Section 3 presents the Posterior Regularization (PR) framework and describes how to encode such constraints in an efficient manner, requiring only repeated inference in the original model to enforce the constraints. Section 4 presents a detailed evaluation of the alignments produced. The constraints over posteriors consistently and significantly outperform the unconstrained HMM model, evaluated against manual annotations. Moreover, this training procedure outperforms the more complex IBM Model 4 nine times out of 12. 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. We also demonstrate a new methodology to avoid overfitting using a small development corpus. Section 5 evaluates the new framework on two different tasks that depend on word alignments. Section 5.1 focuses on MT and shows that the better alignments also lead to better translation systems, adding to similar evidence presented in Ganchev, Graça, and Taskar (2008). Section 5.2 shows that the alignments we produce are better suited for transfer of syntactic dependency parse annotations. An implementation of this work (Graça, Ganchev, and Taskar 2009) is available under a GPL license.¹

1 www.seas.upenn.edu/~strctlrn/CAT/.

2. Background

A 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. 1993b). There are many reasons why a simple word-to-word (1-to-1) correspondence is not possible for every sentence pair: for instance, auxiliary verbs used in one language but not the other (e.g., English *He walked* and French *Il est allé*), articles required in one language but optional in the other (e.g., English *Cars use gas* and Portuguese *Os carros usam gasolina*), cases where the content is expressed using multiple words in one language and a single word in the other language (e.g., agglutination such as English *weapons of mass destruction* and German *Massenvernichtungswaffen*), and expressions translated indirectly. Due to this inherent ambiguity, manual annotations usually distinguish between **sure** correspondences for unambiguous translations, and **possible**, for ambiguous translations (Och and Ney 2003). The top row of Figure 1 shows two word alignments between an English–French sentence pair. We use the following notation: the alignment on the left (right) will be referenced as source–target (target–source) and contains source (target) words as rows and target (source) words as columns. Each entry in the matrix corresponds to a source–target word pair, and is the candidate for an alignment **link**. Sure links are represented as squares with borders, and possible links

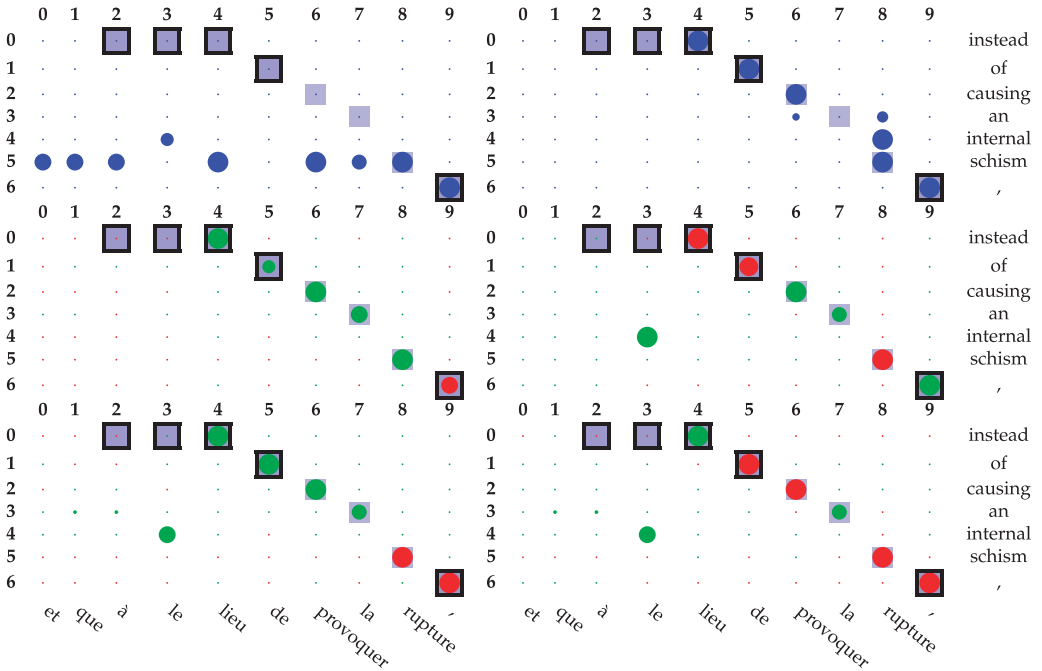


Figure 1 Posterior marginal distributions for different models for an English to French sentence translation. **Left:** EN→FR model. **Right:** FR→EN model. **Top:** Regular HMM posteriors. **Middle:** After applying bijective constraint. **Bottom:** After applying symmetric constraint. Sure alignments 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.

Table 1

Test corpora statistics: English–French, English–Spanish, English–Portuguese, Portuguese–Spanish, Portuguese–French, and Spanish–French.

| Corpus | Sentence Pairs | Ave Length | Max Length | % Sure | % 1-1 |
|--------|----------------|------------|------------|--------|-------|
| En/Fr | 447 | 16/17 | 30/30 | 21 | 98 |
| En/Es | 400 | 29/31 | 90/99 | 67 | 86 |
| En/Pt | 60 | 11/11 | 20/20 | 54 | 91 |
| Pt/Es | 60 | 11/11 | 20/20 | 69 | 92 |
| Pt/Fr | 60 | 11/12 | 20/20 | 77 | 88 |
| Es/Fr | 60 | 11/12 | 20/20 | 79 | 87 |

are represented as squares without borders. Circles indicate the posterior probability associated with a given link and will be explained latter.

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), and the English/Spanish portion of the Europarl corpus (Koehn 2005) where the annotation is from EPPS (Lambert et al. 2005) (En-Es) using standard 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 some of the variety of challenges presented by each corpus. For example, En-Es has longer sentences and hence more ambiguity for alignment. Furthermore, it has a smaller percentage of bijective (1-to-1) alignments, which makes word fertility more important. Overall, the great majority of links are bijective across the corpora (86–98%). This characteristic will be explored by the constraints described in this article. For the evaluations in Section 4, the percentage of sure links (out of all links) will correlate with difficulty because only sure links are considered for recall.

2.1 HMM Word Alignment Model

In this article we focus on the HMM for word alignment proposed by Vogel, Ney, and Tillmann (1996). This model generalizes IBM Models 1 and 2 (Brown et al. 1993b), by introducing a first-order Markov dependence between consecutive alignment link decisions. The model is an (input–output) HMM with I positions whose hidden state sequence $\mathbf{z} = (z_1, \dots, z_I)$ with $z_i \in \{\text{null}, 1, \dots, J\}$ corresponds to a sequence of source word positions, where J is the source sentence length, and with *null* representing unaligned target words. Each observation corresponds to a word in the target language x_i . The probability of an alignment \mathbf{z} and target sentence \mathbf{x} given a source sentence \mathbf{y} can be expressed as:

$$p_{\theta}(\mathbf{x}, \mathbf{z} | \mathbf{y}) = \prod_{i=1}^I p_d(z_i | z_{i-1}) p_t(x_i | y_{z_i}) \quad (1)$$

where $p_t(x_i | y_{z_i})$ is the probability of a target word at position i being a translation of the source word at position z_i (translation probability), and $p_d(z_i | z_{i-1})$ 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). Note that this model is directional: Each target word (observation) can be aligned to at most one source word (hidden state), whereas a source word could be used multiple times.

We refer to translation parameters p_t and distortions parameters p_d jointly as θ . There are several important standard details of the parametrization: The distortion probability $p_d(z_i | z_{i-1})$ depends only on the distance ($z_i - z_{i-1}$) between the source positions the states represent. Only distances in the range ± 5 are modeled explicitly, with larger distances assigned equal probabilities. The probability of the initial hidden state, $p_d(z_1 | z_0)$ is modeled separately from the other distortion probabilities. To incorporate *null* links, we add a translation probability given *null*: $p_t(x_i | y_{null})$. Following standard practice, *null* links also maintain position information and do not allow distortion. To implement this, we create position-specific *null* hidden states for each source position, and set $p_d(null_i | y_{i'}) = 0$ and $p_d(null_i | null_{i'}) = 0$ for all $i \neq i'$. The model is simple, with complexity of inference $O(I \times J^2)$. There are several problems with the model that arise from its directionality, however.

- **Non-bijective:** Multiple target words can be linked to a single source word. This is rarely desirable. For instance, the model produces non-bijective links 22% of the time for En-Fr instead of 2%.
- **Asymmetric:** By switching the (arbitrary) choice of which language is source and which is target, the HMM produces very different results. For example, intersecting the sets of alignments produced by the two possible choices for source preserves less than half of their union for both En-Fr and En-Pt.²

2.2 Training

Standard HMM training seeks model parameters θ that maximize the log-likelihood of the parallel corpus:

$$\text{Log-Likelihood: } \mathcal{L}(\theta) = \hat{\mathbf{E}}[\log p_\theta(\mathbf{x} | \mathbf{y})] = \hat{\mathbf{E}}[\log \sum_{\mathbf{z}} p_\theta(\mathbf{x}, \mathbf{z} | \mathbf{y})] \quad (2)$$

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

$$\text{EM Lower Bound: } \mathcal{L}(\theta) \geq F(q, \theta) = \hat{\mathbf{E}} \left[\sum_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}, \mathbf{y}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z} | \mathbf{y})}{q(\mathbf{z} | \mathbf{x}, \mathbf{y})} \right] \quad (3)$$

² For both of these points, see the experimental setup in Section 4.1.

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

$$\text{E: } q^{t+1}(\mathbf{z} | \mathbf{x}) = \arg \max_{q(\mathbf{z} | \mathbf{x})} F(q, \theta^t) = \arg \min_{q(\mathbf{z} | \mathbf{x})} \text{KL}(q(\mathbf{z} | \mathbf{x}) || p_{\theta^t}(\mathbf{z} | \mathbf{x})) = p_{\theta^t}(\mathbf{z} | \mathbf{x}) \quad (4)$$

$$\text{M: } \theta^{t+1} = \arg \max_{\theta} F(q^{t+1}, \theta) = \arg \max_{\theta} \widehat{\mathbb{E}} \left[\sum_{\mathbf{z}} q^{t+1}(\mathbf{z} | \mathbf{x}) \log p_\theta(\mathbf{x}, \mathbf{z}) \right] \quad (5)$$

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

In Figure 1, each entry in the alignment matrix contains a circle indicating the alignment link posterior for that particular word pair after training an HMM model with the EM algorithm (see the experimental set up in Section 4.1). Note that the link posteriors are concentrated around particular source words (rare words occurring less than five 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 the “garbage collector effect” (Brown et al. 1993a). A rare word in the source language links 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 that 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, because 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 a higher likelihood than correctly linking them or linking them to the special *null* word, because it increases the likelihood of this sentence without lowering the likelihood of many other sentences.

2.3 Decoding

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 is to use Minimum Bayes-Risk (MBR) decoding (Kumar and Byrne 2002; Liang, Taskar, and Klein 2006; Graça, Ganchev, and Taskar 2007). Using this decoding we include an alignment link $i - j$ 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 alignment link. The threshold is tuned on some small amount of labeled data—in our case the development set—to minimize some loss. Kumar and Byrne (2002) study different loss functions that incorporate linguistic knowledge, and show significant

improvement over likelihood decoding. Note that this could potentially result in an alignment having zero probability under the model, as many-to-many alignments can be produced in this way. MBR decoding has several advantages over Viterbi decoding. First, independently of the particular choice of the loss function, by picking a specific threshold we can trade off precision and recall of the predicted word alignments. In fact, in this work when comparing different alignment sets we do not commit to any loss function but instead compare precision vs recall curves, by generating alignments for different thresholds (0..1). Second, with this method we can ignore the *null* word probabilities, which tend to be poorly estimated.

3. Posterior Regularization

Word alignment models in general and the HMM in particular are very gross oversimplifications of the translation process and the optimal likelihood parameters learned often do not correspond to sensible 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. 1993b; 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 propose to use a learning framework called Posterior Regularization (Graça, Ganchev, and Taskar 2007) that incorporates side information into unsupervised estimation in the form of constraints on the model’s posteriors. The constraints are expressed as inequalities on the expected values under the posterior distribution of user-defined constraint features (not necessarily the same features used by the model). Because in most applications what we are interested in are the latent variables (in this case the alignments), constraining the posteriors allows a more direct way to achieve the desired behavior. On the other hand, constraining the expected value of the features instead of adding them to the model allows us to express features that would otherwise make the model intractable. For example, enforcing that each hidden state of an HMM model should be used at most once per sentence would break the Markov property and make the model intractable. In contrast, we will show how to enforce the constraint that each hidden state is used at most once *in expectation*. The underlying model remains unchanged, but the learning method changes. During learning, our method is similar to the EM algorithm with the addition of solving an optimization problem similar to a maximum entropy problem inside the E Step. The following subsections present the Posterior Regularization framework, followed by a description of how to encode two pieces of prior information aimed at solving the problems described at the end of Section 2.

3.1 Posterior Regularization Framework

The goal of the Posterior Regularization (PR) framework is to guide a model during learning towards satisfying some prior knowledge about the desired latent variables (in this case word alignments), encoded as constraints over their expectations. The key advantage of using regularization on posterior expectations is that the base model remains unchanged, but during learning, it is driven to obey the constraints by setting appropriate parameters θ . Moreover, experiments show that enforcing constraints in expectation results in predicted alignments that also satisfy the constraints. More formally, posterior information in PR is specified with sets \mathcal{Q}_x of allowed distributions over the

hidden variables \mathbf{z} which satisfy inequality constraints on some user-defined feature expectations, with violations bounded by $\epsilon \geq 0$:

Constrained Posterior Set:
$$\mathcal{Q}_x = \{q(\mathbf{z} | \mathbf{x}) : \exists \xi, \mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] - \mathbf{b}_x \leq \xi; \|\xi\|_2^2 \leq \epsilon^2\} \quad (6)$$

\mathcal{Q}_x denotes the set of valid distributions where some feature expectations are bounded by \mathbf{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 \mathcal{Q}_x is non-empty for each example x . Furthermore, the set \mathcal{Q}_x needs to be convex. In this work we restrict ourselves to linear inequalities because, as will be shown, subsequently this simplifies the learning algorithm. Note that \mathcal{Q}_x , $\mathbf{f}(\mathbf{x}, \mathbf{z})$, and \mathbf{b}_x also depend on \mathbf{y} , the corresponding source sentence, but we suppress the dependence for brevity. In PR, the log-likelihood of a model is penalized with the KL-divergence between the desired distribution space \mathcal{Q}_x and the model posteriors, $\text{KL}(\mathcal{Q}_x \| p_\theta(\mathbf{z}|\mathbf{x})) = \min_{q(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}_x} \text{KL}(q(\mathbf{z} | \mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x}))$. The regularized objective is:

Posterior Regularized Likelihood:
$$\mathcal{L}(\theta) - \widehat{\mathbf{E}}[\text{KL}(\mathcal{Q}_x \| p_\theta(\mathbf{z}|\mathbf{x}))]. \quad (7)$$

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.

Another way of interpreting the objective is to express the marginal log-likelihood $\mathcal{L}(\theta)$ as a KL distance: $\text{KL}(\delta(\mathbf{x}^n) \| p_\theta(\mathbf{x}))$ where $\delta(\mathbf{x}^n)$ is a delta function at \mathbf{x}^n . Hence the objective is a sum of two average KL terms, one in the space of distributions over \mathbf{x} and one in the space of distributions over \mathbf{z} :

$$-\mathcal{L}(\theta) + \widehat{\mathbf{E}}[\text{KL}(\mathcal{Q}_x \| p_\theta(\mathbf{z}|\mathbf{x}))] = \frac{1}{N} \sum_{n=1}^N \text{KL}(\delta(\mathbf{x}^n) \| p_\theta(\mathbf{x})) + \text{KL}(\mathcal{Q}_{x^n} \| p_\theta(\mathbf{z}|\mathbf{x}^n)) \quad (8)$$

This view of the PR objective is illustrated in Figure 2.

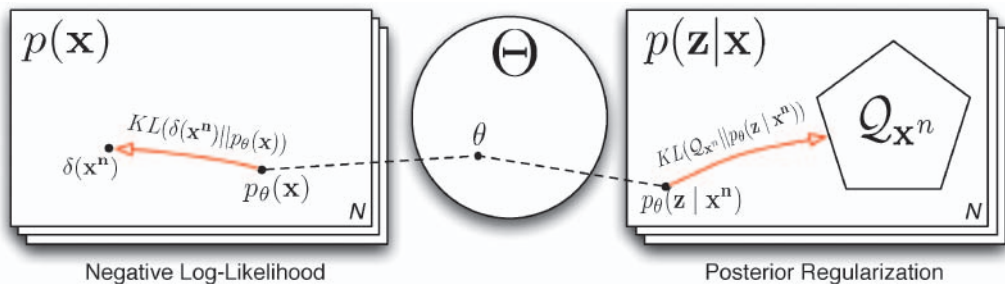


Figure 2 Maximizing the PR objective is equivalent to minimizing the empirical average of two KL divergences: The negative log-likelihood $-\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^N \text{KL}(\delta(\mathbf{x}^n) \| p_\theta(\mathbf{x}))$ plus posterior regularization $\frac{1}{N} \sum_{n=1}^N \text{KL}(\mathcal{Q}_{x^n} \| p_\theta(\mathbf{z}|\mathbf{x}^n))$, where $\delta(\mathbf{x}^n)$ is a delta function at \mathbf{x}^n . The diagram illustrates the effect of the likelihood term and the regularization term operating over the two spaces of distributions: the observed variables \mathbf{x} and the latent variables \mathbf{z} . (The effect of the prior on θ is not shown.)

Computing the PR objective involves solving the optimization problem for each \mathbf{x} :

Primal Projection :
$$\text{KL}(\mathcal{Q}_x \parallel p_\theta(\mathbf{z}|\mathbf{x})) = \min_{q(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}_x} \text{KL}(q(\mathbf{z} | \mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})) \quad (9)$$

Directly minimizing this objective is hard because there is an exponential number of alignments \mathbf{z} ; however, the problem becomes easy to solve in its dual formulation (see Appendix A for derivation):

Dual Projection :
$$\arg \min_{\lambda \geq 0} \mathbf{b}_x^\top \lambda + \log Z(\lambda) + \epsilon \|\lambda\|_2 \quad (10)$$

where $Z(\lambda) = \sum_{\mathbf{z}} p_\theta(\mathbf{z}|\mathbf{x}) \exp(-\lambda \cdot \mathbf{f}(\mathbf{x}, \mathbf{z}))$ is the normalization constant and the primal solution is $q(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x}) \exp\{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\} / Z(\lambda)$. There is one dual variable per expectation constraint, and the dual gradient at $\lambda \neq 0$ is $\nabla(\lambda) = \mathbf{b}_x - \mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] + \epsilon \frac{\lambda_i}{\|\lambda\|_2}$. Note that this primal–dual relationship is very similar to the one between maximum likelihood and maximum entropy. If \mathbf{b}_x corresponds to empirical expectations and $p_\theta(\mathbf{z}|\mathbf{x})$ is uniform, then Equation (10) would be a log-likelihood and Equation (14) (following) would be a maximum entropy problem. As with maximum entropy, gradient computation involves computing an expectation under $q(\mathbf{z} | \mathbf{x})$, which can be performed efficiently if the features $\mathbf{f}(\mathbf{x}, \mathbf{z})$ factor in the same way as the model $p_\theta(\mathbf{x}, \mathbf{z})$, and the constraints are linear. The conditional distribution over \mathbf{z} represented by a graphical model such as HMM can be written as a product of factors over cliques \mathcal{C} :

Factored Posterior:
$$p(\mathbf{z} | \mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi(\mathbf{x}, \mathbf{z}_c) \quad (11)$$

In an HMM, the cliques \mathcal{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 \mathbf{f} is factorized as a sum over the same cliques (we will show below how symmetry and bijectivity constraints can be expressed in this way):

Factored Features:
$$\mathbf{f}(\mathbf{x}, \mathbf{z}) = \sum_{c \in \mathcal{C}} \mathbf{f}(\mathbf{x}, \mathbf{z}_c) \quad (12)$$

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

$$q(\mathbf{z} | \mathbf{x}) = \frac{1}{Z} p(\mathbf{z} | \mathbf{x}) \exp(-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi(\mathbf{x}, \mathbf{z}_c) \exp^{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z}_c)} \quad (13)$$

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 λ .

```

1  $\lambda_i \leftarrow 0;$ 
2 while  $\|\nabla(\lambda)\|_2 > \eta$  do
3    $\phi'(\mathbf{x}, \mathbf{z}_c) \leftarrow \phi(\mathbf{x}, \mathbf{z}_c) \exp^{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z}_c)};$ 
4    $q(\mathbf{z} | \mathbf{x}) \leftarrow \text{forwardBackward}(\phi'(\mathbf{x}, \mathbf{z}_c));$ 
5    $\lambda \leftarrow \lambda + \alpha \beta \nabla(\lambda);$ 
6 end
```

Algorithm 1: Computing $\text{KL}(\mathcal{Q}_x \parallel p_\theta(\mathbf{z}|\mathbf{x})) = \min_{q \in \mathcal{Q}_x} \text{KL}(q(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$

We optimize the dual objective using the gradient based methods shown in Algorithm 1. Here η is an optimization precision, α is a step size chosen with the strong Wolfe’s rule (Nocedal and Wright 1999). Here, $\beta \nabla(\lambda)$ represents an ascent direction chosen as follows: For inequality constraints, it is the projected gradient (Bertsekas 1999); for equality constraints with slack, we use conjugate gradient (Nocedal and Wright 1999), noting that when $\lambda = 0$, the objective is not differentiable. In practice this only happens at the start of optimization and we use a sub-gradient for the first direction.

Computing the projection requires an algorithm for inference in the original model, and uses that inference as a subroutine. For HMM word alignments, we need to make several calls to forward-backward in order to choose λ . Setting the optimization precision η more loosely allows the optimization to terminate more quickly but at a less accurate value. We found that aggressive optimization significantly improves alignment quality for both constraints we used and consequently choose η so that tighter values do not significantly improve performance. This explains why we report better results here in this paper than in Ganchev, Graça, and Taskar (2008), which uses a more naive optimization (see Section 4.1).

3.2 Posterior Regularization via Expectation Maximization

We can optimize the PR objective using a procedure very similar to the expectation maximization (EM) algorithm. Recall from Equation (4) that in the E step, $q(\mathbf{z} | \mathbf{x})$ is set to the posterior over hidden variables given the current θ . To converge to the PR objective, we must modify the E step so that $q(\mathbf{z} | \mathbf{x})$ is a projection of the posteriors onto the constraint set \mathcal{Q}_x for each example \mathbf{x} (Graça, Ganchev, and Taskar 2007).

$$E': \arg \min_{q, \xi} \text{KL}(q(\mathbf{z} | \mathbf{x}) || p_{\theta^t}(\mathbf{z} | \mathbf{x})) \quad \text{s.t.} \quad E_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] - \mathbf{b}_x \leq \xi; \quad \|\xi\|_2^2 \leq \epsilon^2 \quad (14)$$

The new posteriors $q(\mathbf{z} | \mathbf{x})$ are used to compute sufficient statistics for this instance and hence to update the model’s parameters in the M step (Equation (5)), which remains unchanged. This scheme is illustrated in Figure 3 and in Algorithm 2. The only implementation difference is that we must now perform the KL projection before collecting sufficient statistics. We found it can help to also perform this projection at test time, using $q(\mathbf{z} | \mathbf{x}) = \arg \min_{q(\mathbf{z} | \mathbf{x}) \in \mathcal{Q}_x} \text{KL}(q(\mathbf{z} | \mathbf{x}) || p_{\theta}(\mathbf{z} | \mathbf{x}))$ instead of $p_{\theta}(\mathbf{z} | \mathbf{x})$ to decode.

```

1 for  $t = 1..T$  do
2   for each training sentence  $\mathbf{x}$  do
3     E'-Step:  $q^{t+1}(\mathbf{z} | \mathbf{x}) = \arg \min_{q(\mathbf{z} | \mathbf{x}) \in \mathcal{Q}_x} \text{KL}(q(\mathbf{z} | \mathbf{x}) || p_{\theta^t}(\mathbf{z} | \mathbf{x}))$ 
4   end
5   M-Step:  $\theta^{t+1} = \arg \max_{\theta} \widehat{\mathbf{E}} [\sum_{\mathbf{z}} q^{t+1}(\mathbf{z} | \mathbf{x}) \log p_{\theta}(\mathbf{z}, \mathbf{x})]$ 
6 end

```

Algorithm 2: PR optimization via modified EM. E'-Step is computed using Algorithm 1.

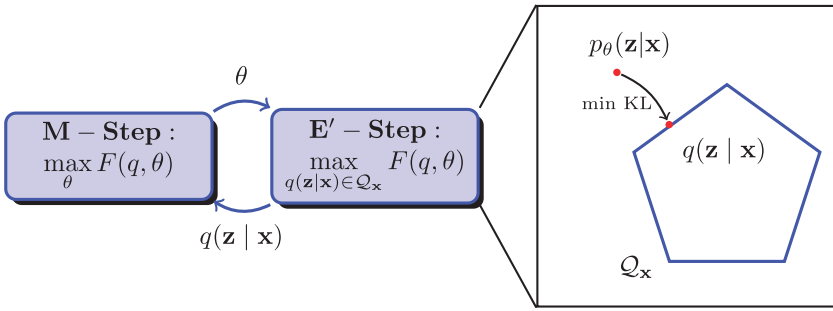


Figure 3
 Modified EM for optimizing PR objective $\mathcal{L}(\theta) - \hat{\mathbb{E}}[\text{KL}(\mathcal{Q}_x \parallel p_\theta(\mathbf{z}|\mathbf{x}))]$.

3.3 Bijectivity Constraints

We observed in Table 1 that most alignments are 1-to-1 and we would like to introduce this prior information into the model. Unfortunately including such a constraint in the model directly breaks the Markov property in a fairly fundamental way. In particular computing the normalization would require the summation of 1-to-1 or near 1-to-1 weighted matchings, which is a classic #P-complete problem. Introducing alignment degree constraints *in expectation* using the PR framework is easy and tractable. We encode them as the constraint $\mathbb{E}[\mathbf{f}(\mathbf{x}, \mathbf{z})] \leq 1$ where we have one feature \mathbf{f} for each source word j that counts how many times it is aligned to a target word in the alignment \mathbf{z} :

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

The second row of Figure 1 shows an example of the posteriors after applying bijectivity constraints; the first row is before the projection. Green (respectively, red) circles indicate that the probability mass for that particular link increased (respectively, decreased) when compared with the EM-trained HMM. For example, in the top left panel, the word *schism* is used more than once, causing erroneous alignments. Projecting to the bijectivity constraint set prevents this and most of the mass is (for this example) moved to the correct word pairs. Enforcing the constraint at training and decoding increases the fraction of 1-to-1 alignment links from 78% to 97.3% for En-Fr (manual annotations have 98.1%); for En-Pt the increase is from 84.7% to 95.8% (manual annotations have 90.8%) (see Section 4.1).

3.4 Symmetry Constraints

The directional nature of the generative models used to recover word alignments conflicts with their interpretation as translations. In practice, we see that the choice of which language is source versus target matters and changes the mistakes made by the model (the first row of panels in 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. Let the directional models be defined as: $\vec{p}(\vec{z})$ (source–target) and $\overleftarrow{p}(\overleftarrow{z})$ (target–source). We suppress dependence on \mathbf{x} and \mathbf{y} for brevity. Define \mathbf{z} to range over the union of all possible

directional alignments $\vec{Z} \cup \overleftarrow{Z}$. We define a mixture model $p(\mathbf{z}) = \frac{1}{2} \overrightarrow{p}(\mathbf{z}) + \frac{1}{2} \overleftarrow{p}(\mathbf{z})$ where $\overleftarrow{p}(\vec{\mathbf{z}}) = 0$ 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 :

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

If the feature f_{ij} has an expected value of zero, then both models predict the i, j link with equal probability. We therefore impose the constraint $E_q[f_{ij}(\mathbf{x}, \mathbf{z})] = 0$ (possibly with some small slack). Note that satisfying this implies satisfying the bijectivity constraint presented earlier. To compute expectations of these features under the model q we only need to be able to compute them under each directional HMM. To see this, we have by the definition of q_λ and p_θ ,

$$q_\lambda(\mathbf{z}|\mathbf{x}) = \frac{\overrightarrow{p}(\mathbf{z}|\mathbf{x}) + \overleftarrow{p}(\mathbf{z}|\mathbf{x}) \exp\{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\}}{2 Z_\lambda} = \frac{\overrightarrow{q}(\mathbf{z}|\mathbf{x}) \frac{Z_{\overrightarrow{q}}}{\overrightarrow{p}(\mathbf{x})} + \overleftarrow{q}(\mathbf{z}|\mathbf{x}) \frac{Z_{\overleftarrow{q}}}{\overleftarrow{p}(\mathbf{x})}}{2 Z_\lambda} \quad (15)$$

where we have defined:

$$\begin{aligned} \overrightarrow{q}(\mathbf{z}|\mathbf{x}) &= \frac{1}{Z_{\overrightarrow{q}}} \overrightarrow{p}(\mathbf{z}, \mathbf{x}) \exp\{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\} \quad \text{with} \quad Z_{\overrightarrow{q}} = \sum_{\mathbf{z}} \overrightarrow{p}(\mathbf{z}, \mathbf{x}) \exp\{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\} \\ \overleftarrow{q}(\mathbf{z}|\mathbf{x}) &= \frac{1}{Z_{\overleftarrow{q}}} \overleftarrow{p}(\mathbf{z}, \mathbf{x}) \exp\{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\} \quad \text{with} \quad Z_{\overleftarrow{q}} = \sum_{\mathbf{z}} \overleftarrow{p}(\mathbf{z}, \mathbf{x}) \exp\{-\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\} \end{aligned}$$

All these quantities can be computed separately in each model using forward–backward and, furthermore, $Z_\lambda = \frac{1}{2}(\frac{Z_{\overrightarrow{q}}}{\overrightarrow{p}(\mathbf{x})} + \frac{Z_{\overleftarrow{q}}}{\overleftarrow{p}(\mathbf{x})})$. The effect of this constraint is illustrated in the bottom panels of Figure 1. The projected link posteriors are equal for the two models, and in most cases the probability mass was moved to the correct alignment links. The exception is the word pair *internal/le*. In this case, the model chose to incorrectly have a high posterior for the alignment link rather than generating *internal* from *null* in one direction and *le* from *null* in the other.

We can measure symmetry of predicted alignments as the ratio of the size of the intersection to the size of the union. Symmetry constraints increase symmetry from 48% to 89.9% for En-Fr and from 48% to 94.2% for En-Pt (see Section 4.1).

4. Alignment Quality Evaluation

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

4.1 Experimental Setup

We discarded all training data sentence pairs where one of the sentences contained more than 40 words. Following common practice, we added the unlabeled development and test data sets to the pool of unlabeled sentences. We initialized the IBM Model 1 translation table with uniform probabilities over word pairs that occur together in the same sentence and trained the IBM Model 1 for 5 iterations. All HMM alignment models

were initialized with the translation table from IBM Model 1 and uniform distortion probabilities. We run each training procedure until the area under the precision/recall curve measured on a development corpus stops increasing (see Figure 4 for an example of such a curve). Using the precision/recall curve gives a broader sense of the model’s performance than using a single point (by tuning a threshold for a particular metric). In most cases this meant four iterations for normal EM training and two iterations using posterior regularization. We suspect that the constraints make the space easier to search.

The convergence criterion for the projection algorithm was the normalized l_2 norm of the gradient (gradient norm divided by number of constraints) being smaller than η (see Algorithm 1). For bijective constraints, we set η to 0.005 and used zero slack. For symmetric constraints, η and slack were set to 0.001. We chose η aggressively and lower values did not significantly increase performance. Less aggressive settings cause degradation of performance: For example, for En-Fr using 10k sentences, and running four iterations of constrained EM, the area under the precision/recall curve for the symmetric model changed from 70% with $\eta = 0.1$ to 85% using $\eta = 0.001$. On the other hand, the number of iterations required to project the constraints increases for smaller values of η . The number of forward-backward calls for normal HMM is 40k (one for each sentence and EM iteration), for the symmetric model using $\eta = 0.1$ was around 41k and using $\eta = 0.001$ was around 26M (14 minutes to 4 hours 14 minutes of training time, 17 times slower, for the different settings of η). We note that better optimization methods, such as L-BFGS, or using a warm start for the parameters at each EM iteration (parameters from the previous iteration), or training the models online, would potentially decrease the running time of our method.

The intent of this experimental section is to evaluate the gains from using constraints during learning, hence the main comparison is between HMM trained with normal EM vs. trained with PR plus constraints. We also report results for IBM Model 4, because it is often used as the default word alignment model, and can be used as a reference. However, we would like to note that IBM Model 4 is a more complex model, able to capture more structure, albeit at the cost of intractable inference. Because our approach is orthogonal to the base model used, the constraints described here could be applied in principle to IBM Model 4 if exact inference was efficient, hopefully yielding similar improvements. We used a standard implementation of IBM Model 4 (Och and Ney 2003) and because changing the existing code is not trivial, we could not use the same stopping criterion to avoid overfitting and we are not able to produce precision/recall curves. We trained IBM Model 4 using the default configuration of the

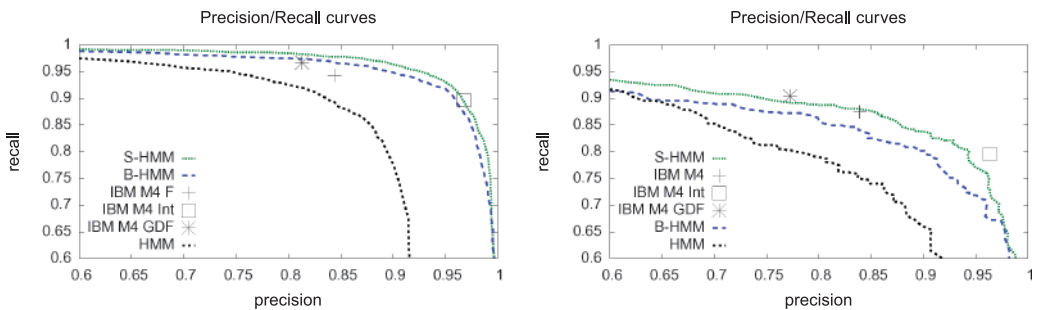


Figure 4

Precision/Recall curves for different models using 1,000k sentences. Precision on the horizontal axis. Left: Hansard EN-FR direction. Right: EN-PT Portuguese-English direction.

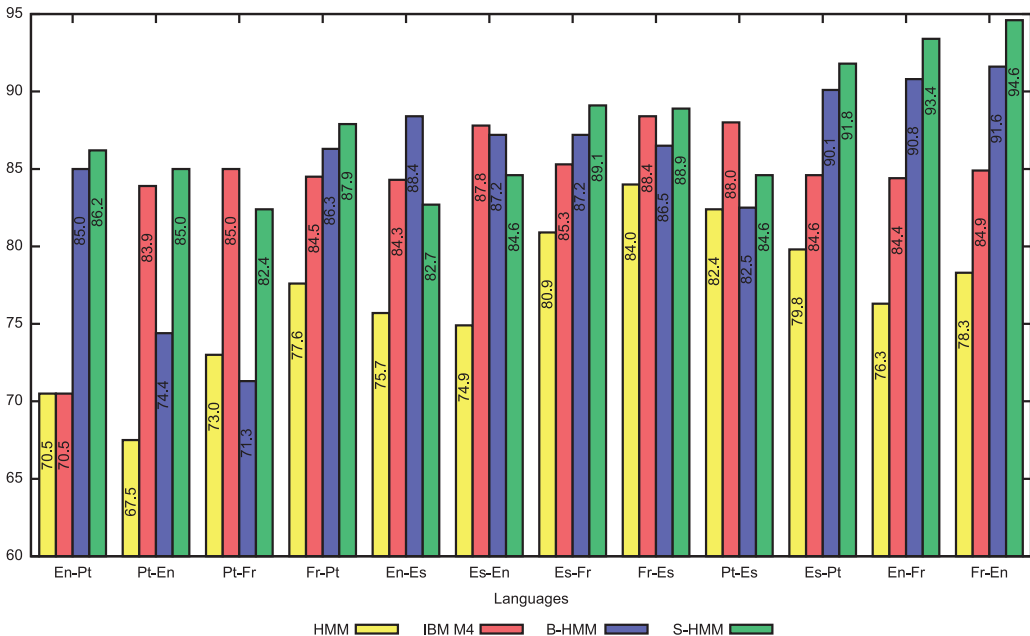


Figure 5 Word alignment precision when the threshold is chosen to achieve IBM Model 4 recall with a difference of ± 0.005 . The average relative increase in precision (against the HMM model) is 10% for IBM Model 4, 11% for B-HMM, and 14% for S-HMM.

MOSES training script.³ This performs five iterations of IBM Model 1, five iterations of HMM, and five iterations of IBM Model 4.

4.2 Alignment Results

In this section we present results on alignment quality. All comparisons are made using MBR decoding because this decoding method always outperforms Viterbi decoding.⁴ For the models with constraints we project the posteriors at decode time (i.e., we use $q(\mathbf{z} | \mathbf{x})$ to decode). This gives a small but consistent improvement. Figure 4 shows precision/recall curves for the different models on the En-Fr corpus using English as the source language (left), and on the En-Pt corpus using Portuguese as the source. Precision/recall curves are obtained by varying the posterior threshold from 0 to 1 and then plotting the different precision and recall values obtained.

We observe several trends from Figure 4. First, both types of constraints improve over the HMM in terms of both precision and recall (their precision/recall curve is always above). Second, S-HMM performs slightly better than B-HMM. IBM Model 4 is comparable with both constraints (after symmetrization). The results for all language pairs are in Figure 5. For ease of comparison, we choose a decoding threshold for HMM models to achieve the recall of the corresponding IBM Model 4 and report precision. Our methods always improve over the HMM by 10% to 15%, and improve over IBM Model 4 nine times out of 12. Comparing the constraints with each other we see that

³ www.statmt.org/moses/?n=FactoredTraining.HomePage.

⁴ IBM Model 4 uses Viterbi decoding as Giza++ does not support MBR decoding.

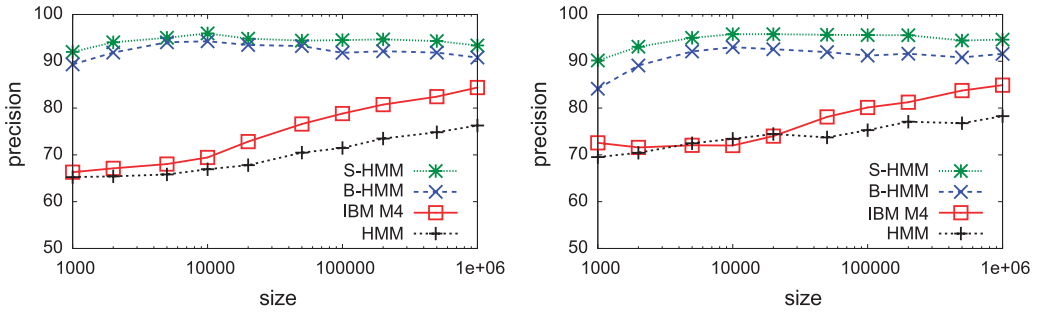


Figure 6 Word alignment precision as a function of training data size (number of sentence pairs). Posterior decoding threshold chosen to achieve IBM Model 4 recall in the Hansard corpus. Right: English as source. Left: French as source.

S-HMM performs better than B-HMM in 10 out of 12 cases. Because S-HMM indirectly enforces bijectivity and models sequential correlations on both sides, this is perhaps not surprising.

Figure 6 shows performance as a function of training data size. As before, we decode to achieve the recall of IBM Model 4. 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%). Greater improvements for small data sizes indicate that our approach can be especially effective for resource-poor language pairs.

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 (Brown et al. 1993a). Figure 7 breaks down performance improvements by common versus rare words. As before, we use posterior decoding, tuning the threshold to match IBM Model 4 recall. For common words, this tuning maintains recall very close for all models so we do not show this in the figure. In the top left panel of Figure 7, we see that precision of common words follows the pattern we saw for the corpus overall: Symmetric and bijective outperform both IBM 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 top right panel of Figure 7 we show the precision for rare words. For the baseline HMM as well as for IBM 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 more accurate, allowing higher recall on the full corpus. In the bottom panel of Figure 7, 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, whereas 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

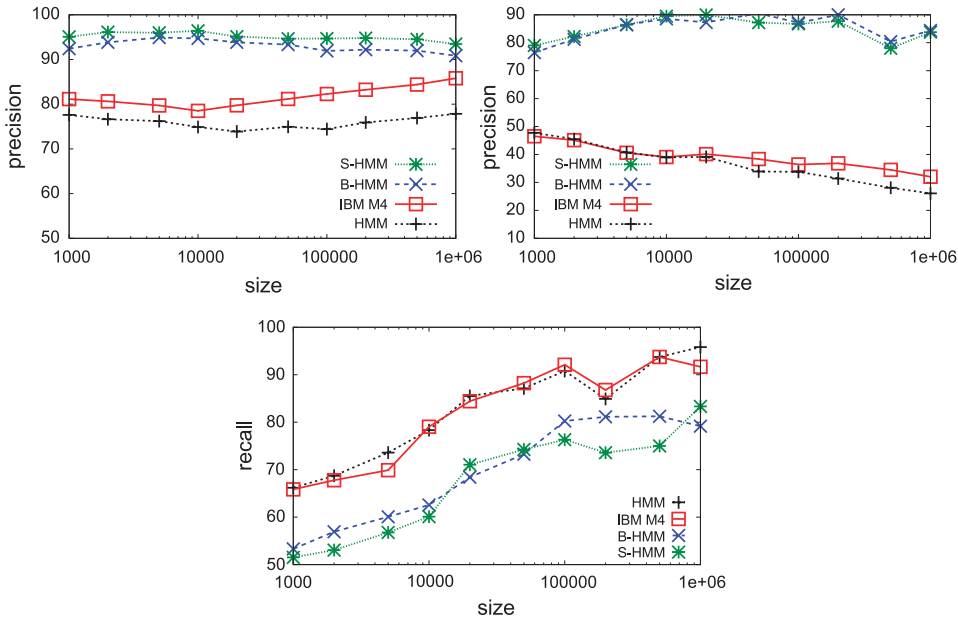


Figure 7
 Precision and Recall as a function of training data size for En-Fr by common and rare words. Top Left: Common Precision, Top Right: Rare Precision, Bottom: Rare Recall.

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, because one wants to have high precision 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. Because they are heuristics tuned for a particular phrase-based translation system, it is not clear when they will help and when they will hinder system performance. In this work we followed a more principled approach that uses

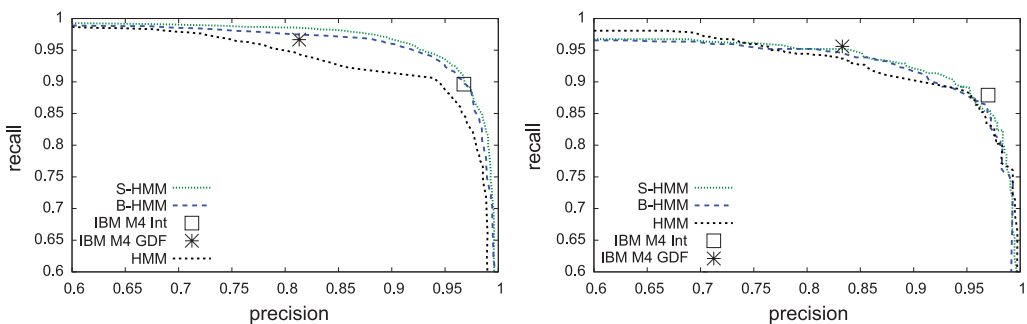


Figure 8
 Precision/recall curves for the different models after soft union symmetrization. Precision is on the horizontal axis. Left EN-FR, Right PT-ES.

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 8 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. This should not be very surprising, because the soft union symmetrization can be viewed as an approximation of our symmetry constraint applied only at decode time. Applying the symmetrization to the model with symmetry constraints does not affect performance.

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-to-1 alignments. Obviously whenever alignments are systematically not 1-to-1, this can lead to errors (for instance the examples described in Section 2).

An example presented in Figure 9 shows the posterior marginal distributions for an English/French sentence pair using the same notation as in Figure 1. In the top panel of

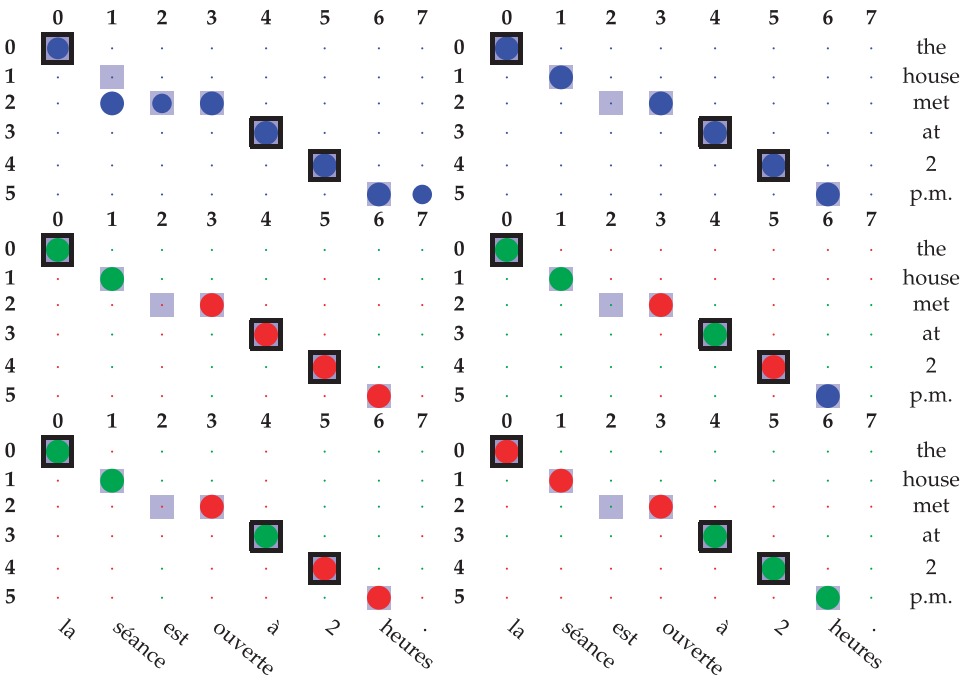


Figure 9 Posterior distributions for different models for an English to French sentence translation. **Left:** EN→FR model. **Right:** FR→ EN model. **Top:** Regular HMM posteriors. **Middle:** After applying the bijective constraint. **Bottom:** After applying the symmetric constraint. Sure alignments 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.

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-to-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 many-to-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. Going back to the right panel of Figure 8, we can see that for low values of precision the HMM model actually achieves better recall than the constraint-based methods. 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. A more complicated model may require approximations that destroy its performance gain, or require larger corpora to estimate its parameters. Another option is to perform some linguistically motivated pre-processing of the language pair to conjoin words. This of course has the disadvantage that it needs to be specific to a language pair in order to include information such as “English simple past 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. Task-Specific Alignment Evaluation

In this section we evaluate the alignments resulting from using the proposed constraints in two different tasks: Statistical machine translation where alignments are used to restrict the number of possible minimal translation units; and syntax transfer, where alignments are used to decide how to transfer dependency links.

5.1 Phrase-Based Machine Translation

We now investigate whether our alignments produce improvements in an end-to-end phrase-based machine translation system. We use a state-of-the-art machine translation system,⁵ and follow the experimental setup used for the 2008 shared task on machine translation (ACL 2008 Third Workshop on Statistical Machine Translation). The full pipeline consists of the following steps: (1) prepare the data (lowercase, tokenize, and filter long sentences); (2) build language models; (3) create word alignments in each direction; (4) symmetrize directional word alignments; (5) build phrase table; (6) tune weights for the phrase table. For more details consult the shared task description.⁶ To evaluate the quality of the produced alignments, we keep the pipeline unchanged, and use the models described earlier to generate the word alignments in Step 3. For Step 4, we use the soft union symmetrization heuristic. Symmetrization has almost no effect on alignments produced by S-HMM, but we use it for uniformity in the experiments. We tested three values of the threshold (0.2, 0.4, 0.6) which try to capture different tradeoffs

⁵ The open source Moses (Hoang et al. 2007) toolkit from www.statmt.org/moses/.

⁶ www.statmt.org/wmt08/baseline.html.

Table 2

BLEU scores for all language pairs. The best threshold was selected according to the development set after the last MERT iteration. Bold denotes the best score.

| | Fr → En | En → Fr | Es → En | En → Es | Pt → En | En → Pt |
|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| IBM M4 GDF | 35.7 | 31.2 | 32.4 | 31.6 | 31.4 | 28.9 |
| HMM SU | 35.9 | 28.9 | 32.3 | 31.6 | 30.9 | 31.6 |
| B-HMM SU | 36.0 | 31.5 | 32.6 | 31.7 | 31.0 | 32.2 |
| S-HMM SU | 35.5 | 31.2 | 31.9 | 32.5 | 31.4 | 32.3 |

of precision vs. recall, and pick the best according to the translation performance on development data. Table 2 summarizes the results for the different corpora. For reference we include IBM Model 4 as suggested in the task description. PR training always outperforms EM training and outperforms IBM Model 4 in all but one experiment. Differences in BLEU range from 0.2 to 0.9. The two constraints help to a different extent for different corpora and translation directions, in a somewhat unpredictable manner. In general our impression is that the connection between alignment quality and BLEU scores is complicated, and changes are difficult to explain and justify. The number of iterations for MERT optimization to converge varied from 2 to 28; and the best choice of threshold on the development set did not always correspond to the best on the test set. Contrary to conventional wisdom in the MT community, bigger phrase tables did not always perform better. In 14 out of 18 cases, the threshold picked was 0.4 (medium size phrase tables) and the other four times 0.2 was picked (smaller phrase tables). When we include only high confidence alignments, more phrases are extracted but many of these are erroneous. Potentially this leads to a poor estimate of the phrase probabilities. See Lopez and Resnik (2006) for further discussion.

5.2 Syntax Transfer

In this section, we compare the different alignments produced with and without PR based on how well they can be used for transfer of linguistic resources across languages. We used the system proposed by Ganchev, Gillenwater, and Taskar (2009). This system uses 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). We consider a parse tree on the source language as a set of dependency edges to be transferred. For each such edge, if both end points are aligned to words in the target language, then the edge is transferred. These edges are then used as weak supervision when training a generative or discriminative dependency parser. In order to evaluate the alignments we computed the fraction of correctly transferred edges as a function of the average number of edges transferred by using supervised parse trees on the target side. By changing the threshold in MBR decoding of alignments, we can trade off accuracy of the transferred edges vs. transferring more edges. We generated supervised parses using the first-order model from the MST parser (McDonald, Crammer, and Pereira 2005) trained on the Penn Treebank for English and the CoNLL X parses for Bulgarian and Spanish. Following Ganchev, Gillenwater, and Taskar (2009), we filter alignment links between words with incompatible POS tags. Figure 10 shows our results for transferring from English to Bulgarian (En→Bg) and from English to Spanish (En→Es). The En→Bg

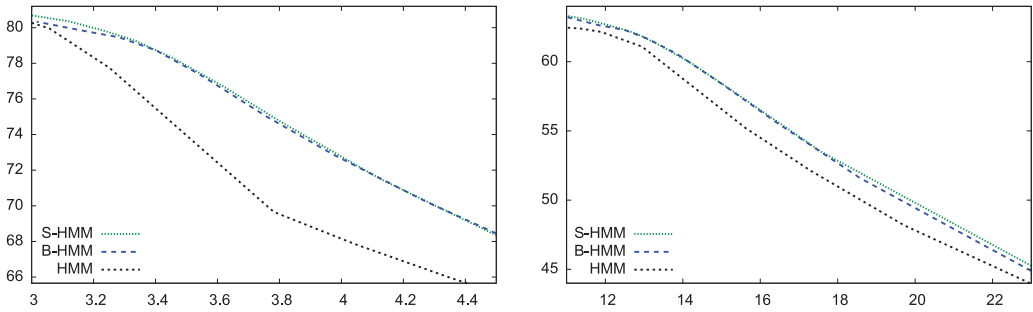


Figure 10

Edge conservation for cross-lingual grammar induction. Left: En→Bg subtitle corpus; Right: En→Es parliamentary proceedings. Vertical axis: percentage of transferred edges that are correct. Horizontal axis: average number of transferred edges per sentence.

results are based on a corpus of movie subtitles (Tiedemann 2007), and are consequently shorter sentences, whereas the En→Es results are based on a corpus of parliamentary proceedings (Koehn 2005). We see in Figure 10 that for both domains, the models trained using posterior regularization perform better than the baseline model trained using EM.

6. Related Work

The idea of introducing constraints over a model to better guide the learning process has appeared before. In the context of word alignment, Deng and Byrne (2005) use a state-duration HMM in order to model word-to-phrase translations. The fertility of each source word is implicitly encoded in the durations of the HMM states. Without any restrictions, likelihood prefers to always use longer phrases and the authors try to control this behavior by multiplying every transition probability by a constant $\eta > 1$. This encourages more transitions and hence shorter phrases. For the task of unsupervised dependency parsing, Smith and Eisner (2006) add a constraint of the form “the average length of dependencies should be X ” to capture the locality of syntax (at least half of the dependencies are between adjacent words), using a scheme they call structural annealing. They modify the model’s distribution over trees $p_\theta(y)$ by a penalty term as: $p'_\theta(y) \propto p_\theta(y)e^{\delta \sum_{e \in y} \text{length}(e)}$, where $\text{length}(e)$ is the surface length of edge e . The factor δ changes from a high value to a lower one so that the preference for short edges (hence a smaller sum) is stronger at the start of training.

These two approaches also have the goal of controlling unsupervised learning, and the form of the modified distributions is reminiscent of the form that the projected posteriors take. However, the approaches differ substantially from PR. Smith and Eisner (2006) make a statement of the form “scale the total length of edges”, which depending on the value of δ will prefer to have more shorter/longer edges. Such statements are not data dependent. Depending on the value of δ , for instance if $\delta \leq 0$, even if the data is such that the model already uses too many short edges on average, this value of δ will push for more short edges. By contrast the statements we can make in PR are of the form “there should be more short edges than long edges”. Such a statement is data-dependent in the sense that if the model satisfies the constraints then we do not need to change it; if it is far from satisfying it we might need to make very dramatic changes.

PR is closely related to the work of Mann and McCallum (2007, 2008), who concurrently developed the idea of using penalties based on posterior expectations of features to guide semi-supervised learning. They call their method generalized expectation (GE) constraints or alternatively expectation regularization. In the original GE framework, the posteriors of the model on unlabeled data are regularized directly. They train a discriminative model, using conditional likelihood on labeled data and an “expectation regularization” penalty term on the unlabeled data:

$$\arg \max_{\theta} \mathcal{L}_{labeled}(\theta) - \lambda \widehat{\mathbf{E}}[\|\mathbf{E}_{p_{\theta}}[\mathbf{f}(\mathbf{x}, \mathbf{z}) - b]\|_2^2]. \quad (16)$$

Notice that there is no intermediate distribution q . For some kinds of constraints this objective is difficult to optimize in θ and in order to improve efficiency, Bellare, Druck, and McCallum (2009) propose interpreting the PR framework as an approximation to the GE objective in Equation (16). They compare the two frameworks on several data sets and find that performance is similar. Liang, Jordan, and Klein (2009) cast the problem of incorporating partial information about latent variables into a Bayesian framework using “measurements,” and after several approximation steps, they arrive at the objective we optimize.

The idea of jointly training two directional models has been explored by Liang, Taskar, and Klein (2006), although under a very different formalization. They define a joint objective $\max_{\theta_1, \theta_2} \widehat{\mathbf{E}} \left[\log \overrightarrow{p}_{\theta_1}(\mathbf{x}) + \log \overleftarrow{p}_{\theta_2}(\mathbf{x}) + \log \sum_{\mathbf{z}} \overrightarrow{p}_{\theta_1}(\mathbf{z} | \mathbf{x}) \overleftarrow{p}_{\theta_2}(\mathbf{z} | \mathbf{x}) \right]$. However, the product distribution $\overrightarrow{p}_{\theta_1}(\mathbf{z} | \mathbf{x}) \overleftarrow{p}_{\theta_2}(\mathbf{z} | \mathbf{x})$ ranges over all one-to-one alignments and computing it is #P-complete (Liang, Taskar, and Klein 2006). They approximate this distribution as a product of marginals: $q(\mathbf{z}) = \prod_{i,j} \overrightarrow{p}_{\theta_1}(\mathbf{z}_{ij} | \mathbf{x}) \overleftarrow{p}_{\theta_2}(\mathbf{z}_{ij} | \mathbf{x})$, but it is not clear what objective the approximate procedure actually optimizes.

7. Conclusion

In this article we explored a novel learning framework, Posterior Regularization, for incorporating rich constraints over the posterior distributions of word alignments. We focused on the HMM word alignment model, and showed how we could incorporate complex constraints like bijectivity and symmetry while keeping the inference in the model tractable. Using these constraints we showed consistent and significant improvements in six different language pairs even when compared to a more complex model such as IBM Model 4. In addition to alleviating the “garbage collector” effect, we show that the obtained posterior distributions better reflect the desired alignments. Both constraints are biasing the models towards 1-to-1 alignments, which may be inappropriate in some situations, and we show some systematic mistakes that the constraints introduce and suggest possible fixes.

We experimented with two different tasks that rely on word alignments, phrase-based MT and syntax transfer. For phrase-based MT, the improved alignments lead to a modest increase in BLEU performance. For syntax transfer, we have shown that the number of edges of a dependency tree that can be accurately transferred from one language to another increases as a result of improved alignments.

Our framework opens up the possibility of efficiently adding many other constraints that are directly applicable to word alignments, such as preferring alignments that respect dependency tree structure, part of speech tags, or syntactic boundaries.

Appendix A: Modified E-Step Dual Derivation

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

$$E': \arg \min_{q(\mathbf{z}|\mathbf{x}), \xi} \text{KL}(q(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}|\mathbf{x})) \quad \text{s.t. } \mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] - \mathbf{b}_x \leq \xi; \|\xi\|_2^2 \leq \epsilon^2.$$

Assuming the set $\mathcal{Q}_x = \{q(\mathbf{z}|\mathbf{x}) : \exists \xi, \mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] - \mathbf{b}_x \leq \xi; \|\xi\|_2^2 \leq \epsilon^2\}$ is non-empty, the corresponding Lagrangian is $\max_{\lambda, \alpha, \gamma} \min_{q(\mathbf{z}|\mathbf{x}), \xi} L(q(\mathbf{z}|\mathbf{x}), \xi, \lambda, \alpha, \gamma)$ with $\lambda \geq 0$ and $\alpha \geq 0$, where

$$\begin{aligned} L(q(\mathbf{z}|\mathbf{x}), \xi, \lambda, \alpha, \gamma) &= \text{KL}(q(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}|\mathbf{x})) + \lambda^{\top} (\mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] - \mathbf{b}_x - \xi) \\ &\quad + \alpha (\|\xi\|_2^2 - \epsilon^2) + \gamma (\sum_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) - 1) \\ \frac{\partial L(q(\mathbf{z}|\mathbf{x}), \xi, \lambda, \alpha, \gamma)}{\partial q(\mathbf{z}|\mathbf{x})} &= \log(q(\mathbf{z}|\mathbf{x})) + 1 - \log(p_{\theta}(\mathbf{z}|\mathbf{x})) + \lambda^{\top} \mathbf{f}(\mathbf{x}, \mathbf{z}) + \gamma = 0 \\ \implies q(\mathbf{z}|\mathbf{x}) &= \frac{p_{\theta}(\mathbf{z}|\mathbf{x}) \exp(-\lambda^{\top} \mathbf{f}(\mathbf{x}, \mathbf{z}))}{e \exp(\gamma)} \\ \frac{\partial L(q(\mathbf{z}|\mathbf{x}), \xi, \lambda, \alpha, \gamma)}{\partial \xi_i} &= 2\alpha \xi_i - \lambda_i = 0 \implies \xi_i = \frac{\lambda_i}{2\alpha} \end{aligned}$$

Plugging $q(\mathbf{z}|\mathbf{x})$ and ξ in $L(q(\mathbf{z}|\mathbf{x}), \xi, \lambda, \alpha, \gamma)$ and taking the derivative with respect to γ :

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

Simplifying $q(\mathbf{z}|\mathbf{x}) = \frac{p_{\theta}(\mathbf{z}|\mathbf{x}) \exp(-\lambda^{\top} \mathbf{f}(\mathbf{x}, \mathbf{z}))}{Z_{\lambda}}$ where $Z_{\lambda} = \sum_{\mathbf{z}} p_{\theta}(\mathbf{z}|\mathbf{x}) \exp(-\lambda^{\top} \mathbf{f}(\mathbf{x}, \mathbf{z}))$ ensures that $q(\mathbf{z}|\mathbf{x})$ is properly normalized. Plugging γ into $L(\lambda, \alpha, \gamma)$ and taking the derivative with respect to α , we get:

$$L(\lambda, \alpha) = -\log(Z_{\lambda}) - \mathbf{b}_x^{\top} \lambda - \frac{\|\lambda\|_2^2}{2\alpha} + \frac{\|\lambda\|_2^2}{4\alpha} - \alpha \epsilon^2 \quad (\text{A.1})$$

$$\frac{\partial L(\lambda, \alpha)}{\partial \alpha} = \frac{\|\lambda\|_2^2}{2\alpha^2} - \frac{\|\lambda\|_2^2}{4\alpha^2} - \epsilon^2 = 0 \implies \alpha = \frac{\|\lambda\|_2}{2\epsilon} \quad (\text{A.2})$$

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

$$\text{Dual } E': \arg \max_{\lambda \geq 0} -\mathbf{b}_x^{\top} \lambda - \log(Z_{\lambda}) - \|\lambda\|_2 \epsilon \quad (\text{A.3})$$

Acknowledgments

J. V. Graça was supported by a fellowship from Fundação para a Ciência e Tecnologia (SFRH/BD/27528/2006) and by FCT project CMU-PT/HuMach/0039/2008. K. Ganchev was partially supported by NSF ITR EIA 0205448. Ben Taskar was partially supported by DARPA CSSG 2009 grant.

References

- Bannard, Colin and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL '05: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 597–604, Morristown, NJ.
- Bellare, Kedar, Gregory Druck, and Andrew McCallum. 2009. Alternating projections

- for learning with expectation constraints. In *Proceedings of the Twenty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–50, Corvallis, OR.
- Bertsekas, Dimitri P. 1999. *Nonlinear Programming: 2nd Edition*. Athena Scientific, Nashua, NH.
- Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, Meredith J. Goldsmith, Jan Hajic, Robert L. Mercer, and Surya Mohanty. 1993a. But dictionaries are data too. In *HLT '93: Proceedings of the Workshop on Human Language Technology*, pages 202–205, Morristown, NJ.
- Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993b. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.
- Chiang, David, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. 2005. The Hiero machine translation system: Extensions, evaluation, and analysis. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 779–786, Vancouver.
- Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Royal Statistical Society, Series B*, 39(1):1–38.
- DeNero, John and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague.
- Deng, Yonggang and William Byrne. 2005. HMM word and phrase alignment for statistical machine translation. In *HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 169–176, Morristown, NJ. Association for Computational Linguistics.
- Fraser, Alexander and Daniel Marcu. 2007. Getting the structure right for word alignment: Leaf. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 51–60, Prague.
- Galley, Michel, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, MA.
- Ganchev, Kuzman, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 369–377, Morristown, NJ.
- Ganchev, Kuzman, João V. Graça, and Ben Taskar. 2008. Better alignments = better translations? In *Proceedings of ACL-08: HLT*, pages 986–993, Columbus, OH.
- Graça, João V., Kuzman Ganchev, and Ben Taskar. 2007. Expectation maximization and posterior constraints. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, pages 569–576.
- Graça, João V., Kuzman Ganchev, and Ben Taskar. 2009. Postcat - posterior constrained alignment toolkit. *The Prague Bulletin Of Mathematical Linguistics - Special Issue: Open Source Tools for Machine Translation*, 91:27–37.
- Graça, João V., Joana P. Pardal, Luísa Coheur, and Diamantino Caseiro. 2008. Building a golden collection of parallel multi-language word alignment. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech.
- Hoang, Hieu, Alexandra Birch, Chris Callison-Burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, and Ondrej Bojar. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague.
- Hwa, Rebecca, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:11–311.
- Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit*, 12–15 September, Phuket.
- Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based

- translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*, pages 48–54, Morristown, NJ.
- Kumar, Shankar and William Byrne. 2002. Minimum Bayes-Risk word alignments of bilingual texts. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, pages 140–147, Philadelphia, PA.
- Lambert, Patrik, Adrià De Gispert, Rafael Banchs, and José B. Marino. 2005. Guidelines for word alignment evaluation and manual alignment. *Language Resources and Evaluation*, 39(4):267–285.
- Liang, Percy, Michael I. Jordan, and Dan Klein. 2009. Learning from measurements in exponential families. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 641–648, New York, NY.
- Liang, Percy, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York, NY.
- Lopez, Adam and Philip Resnik. 2006. Word-based alignment, phrase-based translation: Whats the link? In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA): Visions for the Future of Machine Translation*, pages 90–99, Boston, MA.
- Mann, G. and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the 24th International Conference on Machine Learning*, page 600, Corvallis, OR.
- Mann, Gideon S. and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL-08: HLT*, pages 870–878, Columbus, OH.
- Matusov, Evgeny, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proceedings of the EACL*, pages 33–40, Cambridge.
- McDonald, Ryan, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL '05: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 91–98, Morristown, NJ.
- Neal, Radford M. and Geoffrey 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*. Kluwer, Amsterdam, pages 355–368.
- Nocedal, Jorge and Stephen J. Wright. 1999. *Numerical Optimization*. Springer, Berlin.
- Och, Franz Josef and Hermann Ney. 2000. Improved statistical alignment models. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447, Morristown, NJ.
- Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Smith, Noah A. and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 569–576, Morristown, NJ.
- Snyder, Benjamin and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, OH.
- Tiedemann, Jörg. 2007. Building a multilingual parallel subtitle corpus. In *Proceedings of the 17th Conference on Computational Linguistics in the Netherlands (CLIN 17)*, Leuven.
- Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 836–841, Morristown, NJ.
- Yarowsky, David and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the North American Chapter Of The Association For Computational Linguistics*, pages 1–8, Morristown, NJ.