Sampling Alignment Structure under a Bayesian Translation Model

John DeNero, Alexandre Bouchard-Côté and Dan Klein
Computer Science Department
University of California, Berkeley
{denero, bouchard, klein}@cs.berkeley.edu

Abstract

We describe the first tractable Gibbs sampling procedure for estimating phrase pair frequencies under a probabilistic model of phrase alignment. We propose and evaluate two nonparametric priors that successfully avoid the degenerate behavior noted in previous work, where overly large phrases memorize the training data. Phrase table weights learned under our model yield an increase in BLEU score over the word-alignment based heuristic estimates used regularly in phrase-based translation systems.

1 Introduction

In phrase-based translation, statistical knowledge of translation equivalence is primarily captured by counts of how frequently various phrase pairs occur in training bitexts. Since bitexts do not come segmented and aligned into phrase pairs, these counts are typically gathered by fixing a word alignment and applying phrase extraction heuristics to this word-aligned training corpus. Alternatively, phrase pair frequencies can be learned via a probabilistic model of phrase alignment, but this approach has presented several practical challenges.

In this paper, we address the two most significant challenges in phrase alignment modeling. The first challenge is with inference: computing alignment expectations under general phrase models is #P-hard (DeNero and Klein, 2008). Previous phrase alignment work has sacrificed consistency for efficiency, employing greedy hill-climbing algorithms and constraining inference with word alignments (Marcu and Wong, 2002; DeNero et al., 2006; Birch et al., 2006). We describe a Gibbs sampler that consistently and efficiently approximates expectations, using only polynomial-time computable operators. Despite the combinatorial complexity of the phrase alignment space, our sampled phrase pair expectations are guaranteed to converge to the true posterior distributions under the model (in theory) and do converge to effective values (in practice).

The second challenge in learning phrase alignments is avoiding a degenerate behavior of the general model class: as with many models which can choose between large and small structures, the larger structures win out in maximum likelihood estimation. Indeed, the maximum likelihood estimate of a joint phrase alignment model analyzes each sentence pair as one large phrase with no internal structure (Marcu and Wong, 2002). We describe two nonparametric priors that empirically avoid this degenerate solution.

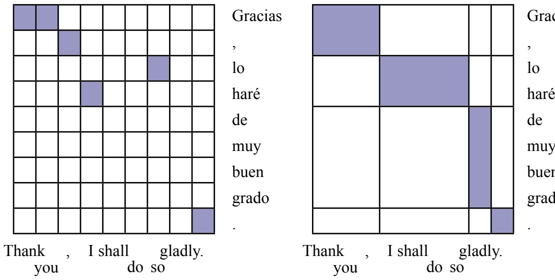
Fixed word alignments are used in virtually every statistical machine translation system, if not to extract phrase pairs or rules directly, then at least to constrain the inference procedure for higher-level models. We estimate phrase translation features consistently using an inference procedure that is not constrained by word alignments, or any other heuristic. Despite this substantial change in approach, we report translation improvements over the standard word-alignment-based heuristic estimates of phrase table weights. We view this result as an important step toward building fully model-based translation systems that rely on fewer procedural heuristics.

2 Phrase Alignment Model

While state-of-the-art phrase-based translation systems include an increasing number of features, translation behavior is largely driven by the phrase pair count ratios $\phi(e|f)$ and $\phi(f|e)$. These features are typically estimated heuristically using the counts $c(\langle e, f\rangle)$ of all phrase pairs in a training corpus that are licensed by word alignments:

$$\phi(e|f) = \frac{c(\langle e, f\rangle)}{\sum_{e'} c(\langle e', f\rangle)} \ .$$

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 314–323,
Honolulu, October 2008. ©2008 Association for Computational Linguistics



(a) example word alignment **(b) example phrase alignment**

Figure 1: In this corpus example, the phrase alignment model found the non-literal translation pair $\langle$*gladly*, *de muy buen grado*$\rangle$ while heuristically-combined word alignment models did not. (a) is a *grow-diag-final-and* combined IBM Model 4 word alignment; (b) is a phrase alignment under our model.

In contrast, a generative model that explicitly aligns pairs of phrases $\langle e, f\rangle$ gives us well-founded alternatives for estimating phrase pair scores. For instance, we could use the model's parameters as translation features. In this paper, we compute the expected counts of phrase pairs in the training data according to our model, and derive features from these expected counts. This approach endows phrase pair scores with well-defined semantics relative to a probabilistic model. Practically, phrase models can discover high-quality phrase pairs that often elude heuristics, as in Figure 1. In addition, the model-based approach fits neatly into the framework of statistical learning theory for unsupervised problems.

2.1 Generative Model Description

We first describe the symmetric joint model of Marcu and Wong (2002), which we will extend. A two-step generative process constructs an ordered set of English phrases $e_{1:m}$, an ordered set of foreign phrases $f_{1:n}$, and a phrase-to-phrase alignment between them, $\mathbf{a} = \{(j, k)\}$ indicating that $\langle e_j, f_k\rangle$ is an aligned pair.

1. Choose a number of components ℓ and generate each of ℓ phrase pairs independently.

2. Choose an ordering for the phrases in the foreign language; the ordering for English is fixed by the generation order.[1]

In this process, $m = n = |\mathbf{a}|$; all phrases in both sentences are aligned one-to-one.

We parameterize the choice of ℓ using a geometric distribution, denoted P_{G}, with stop parameter $p_{\$}$:

$$P(\ell) = P_{\mathrm{G}}(\ell; p_{\$}) = p_{\$} \cdot (1 - p_{\$})^{\ell-1} \ .$$

Each aligned phrase pair $\langle e, f\rangle$ is drawn from a multinomial distribution θ_{J} which is unknown. We fix a simple distortion model, setting the probability of a permutation of the foreign phrases proportional to the product of position-based distortion penalties for each phrase:

$$\begin{aligned} P(\mathbf{a}|\{\langle e, f\rangle\}) &\propto \prod_{a \in \mathbf{a}} \delta(a) \\ \delta(a = (j, k)) &= b^{|pos(e_j) - pos(f_k) \cdot s|} \ , \end{aligned}$$

where $pos(\cdot)$ denotes the word position of the start of a phrase, and s the ratio of the length of the English to the length of the foreign sentence. This positional distortion model was deemed to work best by Marcu and Wong (2002).

We can now state the joint probability for a phrase-aligned sentence consisting of ℓ phrase pairs:

$$P(\{\langle e, f\rangle\}, \mathbf{a}) = P_{\mathrm{G}}(\ell; p_{\$}) P(\mathbf{a}|\{\langle e, f\rangle\}) \prod_{\langle e, f\rangle} \theta_{\mathrm{J}}(\langle e, f\rangle) \ .$$

While this model has several free parameters in addition to θ_{J}, we fix them to reasonable values to focus learning on the phrase pair distribution.[2]

2.2 Unaligned Phrases

Sentence pairs do not always contain equal information on both sides, and so we revise the generative story to include unaligned phrases in both sentences. When generating each component of a sentence pair, we first decide whether to generate an aligned phrase pair or, with probability $p_{\o}$, an unaligned phrase.[3] Then, we either generate an aligned phrase pair from θ_{J} or an unaligned phrase from θ_{N}, where θ_{N} is a multinomial over phrases. Now, when generating $e_{1:m}$, $f_{1:n}$ and alignment $\mathbf{a}$, the number of phrases $m + n$ can be greater than $2 \cdot |\mathbf{a}|$.

[1] We choose the foreign to reorder without loss of generality.

[2] Parameters were chosen by hand during development on a small training corpus. $p_{\$} = 0.1$, $b = 0.85$ in experiments.

[3] We strongly discouraged unaligned phrases in order to align as much of the corpus as possible: $p_{\o} = 10^{-10}$ in experiments.

To unify notation, we denote unaligned phrases as phrase pairs with one side equal to *null*: $\langle e, null\rangle$ or $\langle null, f\rangle$. Then, the revised model takes the form:

$$\begin{aligned} P(\{\langle e,f\rangle\},\mathbf{a}) &= P_{\mathrm{G}}(\ell; p_{\$})P(\mathbf{a}|\{\langle e,f\rangle\})\prod_{\langle e,f\rangle} P_{\mathrm{M}}(\langle e,f\rangle) \\ P_{\mathrm{M}}(\langle e,f\rangle) &= p_{\o}\theta_{\mathrm{N}}(\langle e,f\rangle) + (1-p_{\o})\theta_{\mathrm{J}}(\langle e,f\rangle)\,. \end{aligned}$$

In this definition, the distribution θ_{N} gives non-zero weight only to unaligned phrases of the form $\langle e, null\rangle$ or $\langle null, f\rangle$, while θ_{J} gives non-zero weight only to aligned phrase pairs.

3 Model Training and Expectations

Our model involves observed sentence pairs, which in aggregate we can call x, latent phrase segmentations and alignments, which we can call z, and parameters θ_{J} and θ_{N}, which together we can call θ. A model such as ours could be used either for the learning of the key phrase pair parameters in θ, or to compute expected counts of phrase pairs in our data. These two uses are very closely related, but we focus on the computation of phrase pair expectations. For exposition purposes, we describe a Gibbs sampling algorithm for computing expected counts of phrases under $P(z|x,\theta)$ for fixed θ. Such expectations would be used, for example, to compute maximum likelihood estimates in the E-step of EM. In Section 4, we instead compute expectations under $P(z|x)$, with θ marginalized out entirely.

In a Gibbs sampler, we start with a *complete* phrase segmentation and alignment, state z_0, which sets all latent variables to some initial configuration. We then produce a sequence of sample states z_i, each of which differs from the last by some small local change. The samples z_i are guaranteed (in the limit) to consistently approximate the conditional distribution $P(z|x,\theta)$ (or $P(z|x)$ later). Therefore, the average counts of phrase pairs in the samples converge to expected counts under the model. Normalizing these expected counts yields estimates for the features $\phi(e|f)$ and $\phi(f|e)$.

Gibbs sampling is not new to the natural language processing community (Teh, 2006; Johnson et al., 2007). However, it is usually used as a search procedure akin to simulated annealing, rather than for approximating expectations (Goldwater et al., 2006; Finkel et al., 2007). Our application is also atypical for an NLP application in that we use an approximate sampler not only to include Bayesian prior information (section 4), but also because computing phrase alignment expectations exactly is a #P-hard problem (DeNero and Klein, 2008). That is, we could not run EM exactly, even if we wanted maximum likelihood estimates.

3.1 Related Work

Expected phrase pair counts under $P(z|x,\theta)$ have been approximated before in order to run EM. Marcu and Wong (2002) employed local search from a heuristic initialization and collected alignment counts during a hill climb through the alignment space. DeNero et al. (2006) instead proposed an exponential-time dynamic program pruned using word alignments. Subsequent work has relied heavily on word alignments to constrain inference, even under reordering models that admit polynomial-time E-steps (Cherry and Lin, 2007; Zhang et al., 2008).

None of these approximations are consistent, and they offer no method of measuring their biases. Gibbs sampling is not only consistent in the limit, but also allows us to add Bayesian priors conveniently (section 4). Of course, sampling has liabilities as well: we do not know in advance how long we need to run the sampler to approximate the desired expectations "closely enough."

Snyder and Barzilay (2008) describe a Gibbs sampler for a bilingual morphology model very similar in structure to ours. However, the basic sampling step they propose – resampling all segmentations and alignments for a sequence at once – requires a #P-hard computation. While this asymptotic complexity was apparently not prohibitive in the case of morphological alignment, where the sequences are short, it is prohibitive in phrase alignment, where the sentences are often very long.

3.2 Sampling with the SWAP Operator

Our Gibbs sampler repeatedly applies each of five operators to each position in each training sentence pair. Each operator freezes all of the current state z_i except a small local region, determines all the ways that region can be reconfigured, and then chooses a (possibly) slightly different z_{i+1} from among those outcomes according to the conditional probability of each, given the frozen remainder of the state. This frozen region of the state is called a *Markov blanket* (denoted m), and plays a critical role in proving the correctness of the sampler.

The first operator we consider is SWAP, which changes alignments but not segmentations. It freezes the set of phrases, then picks two English phrases e_1 and e_2 (or two foreign phrases, but we focus on the English case). All alignments are frozen except the phrase pairs $\langle e_1, f_1 \rangle$ and $\langle e_2, f_2 \rangle$. SWAP chooses between keeping $\langle e_1, f_1 \rangle$ and $\langle e_2, f_2 \rangle$ aligned as they are (outcome o_0), or swapping their alignments to create $\langle e_1, f_2 \rangle$ and $\langle e_2, f_1 \rangle$ (outcome o_1).

SWAP chooses stochastically in proportion to each outcome's posterior probability: $P(o_0|m, x, \theta)$ and $P(o_1|m, x, \theta)$. Each phrase pair in each outcome contributes to these posteriors the probability of adding a new pair, deciding whether it is null, and generating the phrase pair along with its contribution to the distortion probability. This is all captured in a succinct potential function $\psi(\langle e, f \rangle) =$

$$\begin{cases} (1-p_{\$})\;(1-p_{\emptyset})\;\theta_{\mathrm{J}}(\langle e, f \rangle)\;\delta(\langle e, f \rangle) & e \;\&\; f \text{ non-}\mathit{null} \\ (1-p_{\$}) \cdot p_{\emptyset} \cdot \theta_{\mathrm{N}}(\langle e, f \rangle) & \text{otherwise} \end{cases}$$

.

Thus, outcome o_0 is chosen with probability $P(o_0|m, x, \theta) =$

$$\frac{\psi(\langle e_1, f_1 \rangle)\psi(\langle e_2, f_2 \rangle)}{\psi(\langle e_1, f_1 \rangle)\psi(\langle e_2, f_2 \rangle) + \psi(\langle e_1, f_2 \rangle)\psi(\langle e_2, f_1 \rangle)}.$$

Operators in a Gibbs sampler require certain conditions to guarantee the correctness of the sampler. First, they must choose among *all possible configurations* of the unfrozen local state. Second, immediately re-applying the operator from any outcome must yield the same set of outcome options as before.[4] If these conditions are not met, the sampler may no longer be guaranteed to yield consistent approximations of the posterior distribution.

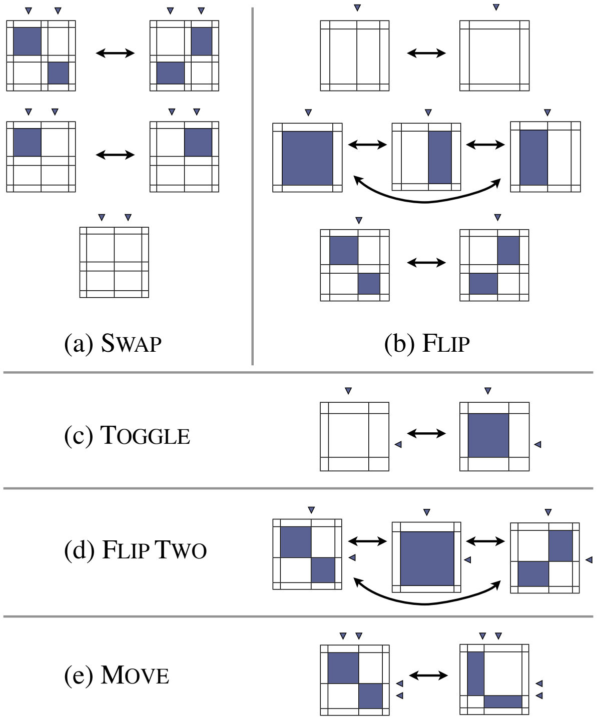


Figure 2: Each local operator manipulates a small portion of a single alignment. Relevant phrases are exaggerated for clarity. The outcome sets (depicted by arrows) of each possible configuration are fully connected. Certain configurations cannot be altered by certain operators, such as the final configuration in SWAP. Unalterable configurations for TOGGLE have been omitted for space.

A subtle issue arises with SWAP as defined: should it also consider an outcome o_2 of $\langle e_1, \mathit{null} \rangle$ and $\langle e_2, \mathit{null} \rangle$ that removes alignments? No part of the frozen state is changed by removing these alignments, so the first Gibbs condition dictates that we must include o_2. However, after choosing o_2, when we reapply the operator to positions e_1 and e_2, we freeze all alignments except $\langle e_1, \mathit{null} \rangle$ and $\langle e_2, \mathit{null} \rangle$, which prevents us from returning to o_0. Thus, we fail to satisfy the second condition. This point is worth emphasizing because some prior work has treated Gibbs sampling as randomized search and, intentionally or otherwise, proposed inconsistent operators.

Luckily, the problem is not with SWAP, but with our justification of it: we can salvage SWAP by augmenting its Markov blanket. Given that we have selected $\langle e_1, f_1 \rangle$ and $\langle e_2, f_2 \rangle$, we not only freeze all other alignments and phrase boundaries, but also the number of aligned phrase pairs. With this count held invariant, o_2 is not among the possible outcomes of SWAP given m. Moreover, regardless of the outcome chosen, SWAP can immediately be reapplied at the same location with the same set of outcomes.

All the possible starting configurations and outcome sets for SWAP appear in Figure 2(a).

[4] These are two sufficient conditions to guarantee that the Metropolis-Hastings acceptance ratio of the sampling step is 1.

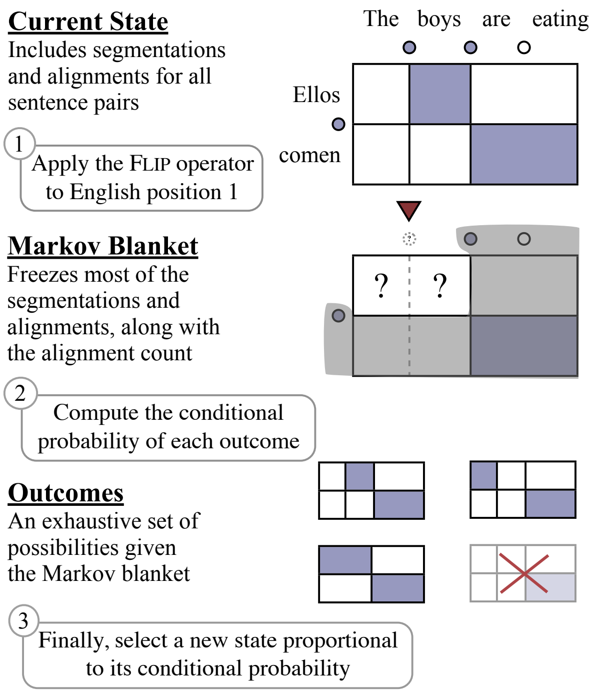

Current State
Includes segmentations and alignments for all sentence pairs
1
Apply the FLIP operator to English position 1
The
boys
are
eating
Ellos
comen
Markov Blanket
Freezes most of the segmentations and alignments, along with the alignment count
?
?
2
Compute the conditional probability of each outcome
Outcomes
An exhaustive set of possibilities given the Markov blanket
3
Finally, select a new state proportional to its conditional probability

Figure 3: The three steps involved in applying the FLIP operator. The Markov blanket freezes all segmentations except English position 1 and all alignments except those for *Ellos* and *The boys*. The blanket also freezes the number of alignments, which disallows the lower right outcome.

3.3 The FLIP operator

SWAP can arbitrarily shuffle alignments, but we need a second operator to change the actual phrase boundaries. The FLIP operator changes the status of a single *segmentation position*[5] to be either a phrase boundary or not. In this sense FLIP is a bilingual analog of the segmentation boundary flipping operator of Goldwater et al. (2006).

Figure 3 diagrams the operator and its Markov blanket. First, FLIP chooses any between-word position in either sentence. The outcome sets for FLIP vary based on the current segmentation and adjacent alignments, and are depicted in Figure 2.

Again, for FLIP to satisfy the Gibbs conditions, we must augment its Markov blanket to freeze not only all other segmentation points and alignments, but also the number of aligned phrase pairs. Otherwise, we end up allowing outcomes from which we cannot return to the original state by reapplying FLIP. Consequently, when a position is already segmented and both adjacent phrases are currently aligned, FLIP cannot unsegment the point because it can't create two aligned phrase pairs with the one larger phrase that results (see bottom of Figure 2(b)).

3.4 The TOGGLE operator

Both SWAP and FLIP freeze the number of alignments in a sentence. The TOGGLE operator, on the other hand, can add or remove individual alignment links. In TOGGLE, we first choose an e_1 and f_1. If $\langle e_1, f_1 \rangle \in \mathbf{a}$ or both e_1 and f_1 are *null*, we freeze all segmentations and the rest of the alignments, and choose between including $\langle e_1, f_1 \rangle$ in the alignment or leaving both e_1 and f_1 unaligned. If only one of e_1 and f_1 are aligned, or they are not aligned to each other, then TOGGLE does nothing.

3.5 A Complete Sampler

Together, FLIP, SWAP and TOGGLE constitute a complete Gibbs sampler that consistently samples from the posterior $P(z|x, \theta)$. Not only are these operators valid Gibbs steps, but they also can form a path of positive probability from any source state to any target state in the space of phrase alignments (formally, the induced Markov chain is *irreducible*). Such a path can at worst be constructed by unaligning all phrases in the source state with TOGGLE, composing applications of FLIP to match the target phrase boundaries, then applying TOGGLE to match the target alignments.

We include two more local operators to speed up the rate at which the sampler explores the hypothesis space. In short, FLIP TWO simultaneously flips an English and a foreign segmentation point (to make a large phrase out of two smaller ones or vice versa), while MOVE shifts an aligned phrase boundary to the left or right. We omit details for lack of space.

3.6 Phrase Pair Count Estimation

With our sampling procedure in place, we can now estimate the expected number of times a given phrase pair occurs in our data, for fixed θ, using a Monte-Carlo average,

$$\frac{1}{N} \sum_{i=1}^{N} \text{count}_{\langle e,f \rangle}(x, z_i) \xrightarrow{a.s.} \mathbb{E}\left[\text{count}_{\langle e,f \rangle}(x, \cdot)\right] \quad .$$

[5] A segmentation position is a position between two words that is also potentially a boundary between two phrases in an aligned sentence pair.

The left hand side is simple to compute; we count aligned phrase pairs in each sample we generate. In practice, we only count phrase pairs after applying every operator to every position in every sentence (one iteration).[6] Appropriate normalizations of these expected counts can be used either in an M-step as maximum likelihood estimates, or to compute values for features $\phi(f|e)$ and $\phi(e|f)$.

4 Nonparametric Bayesian Priors

The Gibbs sampler we presented addresses the inference challenges of learning phrase alignment models. With slight modifications, it also enables us to include prior information into the model. In this section, we treat θ as a random variable and shape its prior distribution in order to correct the well-known degenerate behavior of the model.

4.1 Model Degeneracy

The structure of our joint model penalizes explanations that use many small phrase pairs. Each phrase pair token incurs the additional expense of generation and distortion. In fact, the maximum likelihood estimate of the model puts mass on $\langle e, f\rangle$ pairs that span entire sentences, explaining the training corpus with one phrase pair per sentence.

Previous phrase alignment work has primarily mitigated this tendency by constraining the *inference procedure*, for example with word alignments and linguistic features (Birch et al., 2006), or by disallowing large phrase pairs using a non-compositional constraint (Cherry and Lin, 2007; Zhang et al., 2008). However, the problem lies with the model, and therefore should be corrected in the model, rather than the inference procedure.

Model-based solutions appear in the literature as well, though typically combined with word alignment constraints on inference. A sparse Dirichlet prior coupled with variational EM was explored by Zhang et al. (2008), but it did not avoid the degenerate solution. Moore and Quirk (2007) proposed a new conditional model structure that does not cause large and small phrases to compete for probability mass. May and Knight (2007) added additional model terms to balance the cost of long and short derivations in a syntactic alignment model.

4.2 A Dirichlet Process Prior

We control this degenerate behavior by placing a Dirichlet process (DP) prior over θ_{J}, the distribution over aligned phrase pairs (Ferguson, 1973).

If we were to assume a maximum number K of phrase pair types, a (finite) Dirichlet distribution would be an appropriate prior. A draw from a K-dimensional Dirichlet distribution is a list of K real numbers in $[0, 1]$ that sum to one, which can be interpreted as a distribution over K phrase pair types.

However, since the event space of possible phrase pairs is in principle unbounded, we instead use a Dirichlet process. A draw from a DP is a *countably infinite* list of real numbers in $[0, 1]$ that sum to one, which we interpret as a distribution over a countably infinite list of phrase pair types.[7]

The Dirichlet distribution and the DP distribution have similar parameterizations. A K-dimensional Dirichlet can be parameterized with a *concentration parameter* $\alpha > 0$ and a *base distribution* $M_0 = (\mu_1, \dots, \mu_{K-1})$, with $\mu_i \in (0, 1)$.[8] This parameterization has an intuitive interpretation: under these parameters, the average of independent samples from the Dirichlet will converge to M_0. That is, the average of the ith element of the samples will converge to μ_i. Hence, the base distribution M_0 characterizes the sample mean. The concentration parameter α only affects the variance of the draws.

Similarly, we can parameterize the Dirichlet process with a concentration parameter α (that affects only the variance) and a base distribution M_0 that determines the mean of the samples. Just as in the finite Dirichlet case, M_0 is simply a probability distribution, but now with countably infinite support: all possible phrase pairs in our case. In practice, we can use an unnormalized M_0 (a base measure) by appropriately rescaling α.

In our model, we select a base measure that strongly prefers shorter phrases, encouraging the model to use large phrases only when it has sufficient evidence for them. We continue the model:

[6] For experiments, we ran the sampler for 100 iterations.

[7] Technical note: to simplify exposition, we restrict the discussion to settings such as ours where the base measure of the DP has countable support.

[8] This parametrization is equivalent to the standard pseudo-counts parametrization of K positive real numbers. The bijection is given by $\alpha = \sum_{i=1}^{K} \tilde{\alpha}_i$ and $\mu_i = \tilde{\alpha}_i/\alpha$, where $(\tilde{\alpha}_1, \dots, \tilde{\alpha}_K)$ are the pseudo-counts.

$$
\begin{aligned}
\theta_{\mathrm{J}} &\sim DP(M_0, \alpha) \\
M_0(\langle e, f\rangle) &= [P_f(f)P_{\mathrm{WA}}(e|f) \cdot P_e(e)P_{\mathrm{WA}}(f|e)]^{\frac{1}{2}} \\
P_f(f) &= P_{\mathrm{G}}(|f|; p_s) \cdot \left(\frac{1}{n_f}\right)^{|f|} \\
P_e(e) &= P_{\mathrm{G}}(|e|; p_s) \cdot \left(\frac{1}{n_e}\right)^{|e|} .
\end{aligned}
$$

.

P_{WA} is the IBM model 1 likelihood of one phrase conditioned on the other (Brown et al., 1994). P_f and P_e are uniform over types for each phrase length: the constants n_f and n_e denote the vocabulary size of the foreign and English languages, respectively, and P_{G} is a geometric distribution.

Above, θ_{J} is drawn from a DP centered on the geometric mean of two joint distributions over phrase pairs, each of which is composed of a monolingual unigram model and a lexical translation component. This prior has two advantages. First, we pressure the model to use smaller phrases by increasing p_s ($p_s = 0.8$ in experiments). Second, we encourage good phrase pairs by incorporating IBM Model 1 distributions. This use of word alignment distributions is notably different from lexical weighting or word alignment constraints: we are supplying prior knowledge that phrases will generally follow word alignments, though with enough corpus evidence they need not (and often do not) do so in the posterior samples. The model proved largely insensitive to changes in the sparsity parameter α, which we set to 100 for experiments.

4.3 Unaligned phrases and the DP Prior

Introducing unaligned phrases invites further degenerate megaphrase behavior: a sentence pair can be generated cheaply as two unaligned phrases that each span an entire sentence. We attempted to place a similar DP prior over θ_{N}, but surprisingly, this modeling choice invoked yet another degenerate behavior. The DP prior imposes a *rich-get-richer* property over the phrase pair distribution, strongly encouraging the model to reuse existing pairs rather than generate new ones. As a result, common words consistently aligned to *null*, even while suitable translations were present, simply because each null alignment reinforced the next. For instance, *the* was always unaligned.

Instead, we fix θ_{N} to a simple unigram model that is uniform over word types. This way, we discourage unaligned phrases while focusing learning on θ_{J}. For simplicity, we reuse $P_f(f)$ and $P_e(e)$ from the prior over θ_{J}.

$$
\theta_{\mathrm{N}}(\langle e, f\rangle) = \begin{cases} \frac{1}{2} \cdot P_e(e) & \text{if } f = \mathit{null} \\ \frac{1}{2} \cdot P_f(f) & \text{if } e = \mathit{null} \end{cases} .
$$

The $\frac{1}{2}$ represents a choice of whether the aligned phrase is in the foreign or English sentence.

4.4 Collapsed Sampling with a DP Prior

Our entire model now has the general form $P(x, z, \theta_{\mathrm{J}})$; all other model parameters have been fixed. Instead of searching for a suitable θ_{J},[9] we sample from the posterior distribution $P(z|x)$ with θ_{J} marginalized out.

To this end, we convert our Gibbs sampler into a collapsed Gibbs sampler[10] using the Chinese Restaurant Process (CRP) representation of the DP (Aldous, 1985). With the CRP, we avoid the problem of explicitely representing samples from the DP. CRP-based samplers have served the community well in related language tasks, such as word segmentation and coreference resolution (Goldwater et al., 2006; Haghighi and Klein, 2007).

Under this representation, the probability of each sampling outcome is a simple expression in terms of the state of the rest of the training corpus (the Markov blanket), rather than explicitly using θ_{J}.

Let z_m be the set of aligned phrase pair tokens observed in the rest of the corpus. Then, when $\langle e, f\rangle$ is aligned (that is, neither e nor f are *null*), the conditional probability for a pair $\langle e, f\rangle$ takes the form:

$$
\tau(\langle e, f\rangle | z_m) = \frac{\text{count}_{\langle e,f\rangle}(z_m) + \alpha \cdot M_0(\langle e, f\rangle)}{|z_m| + \alpha},
$$

where $\text{count}_{\langle e,f\rangle}(z_m)$ is the number of times that $\langle e, f\rangle$ appears in z_m. We can write this expression thanks to the exchangeability of the model. For further exposition of this collapsed sampler posterior,

[9] For instance, using approximate MAP EM.

[10] A collapsed sampler is simply one in which the model parameters have been marginalized out.

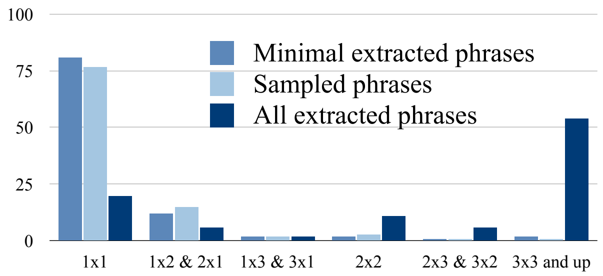


Figure 4: The distribution of phrase pair sizes (denoted *English length* x *foreign length*) favors small phrases under the model.

see Goldwater et al. (2006).[11]

The sampler remains exactly the same as described in Section 3, except that the posterior conditional probability of each outcome uses a revised potential function $\psi_{\mathrm{DP}}(\langle e, f\rangle) =$

$$\begin{cases} (1-p_{\$})\ (1-p_{\emptyset})\ \tau(\langle e,f\rangle)\ \delta(\langle e,f\rangle) & e\ \&\ f \text{ non-}\mathit{null} \\ (1-p_{\$}) \cdot p_{\emptyset} \cdot \theta_{\mathrm{N}}(\langle e,f\rangle) & \text{otherwise} \ . \end{cases}$$

ψ_{DP} is like ψ, but the fixed θ_{J} is replaced with the constantly-updated τ function.

4.5 Degeneracy Analysis

Figure 4 shows a histogram of phrase pair sizes in the distribution of expected counts under the model. As reference, we show the size distribution of both *minimal* and *all* phrase pairs extracted from word alignments using the standard heuristic. Our model tends to select minimal phrases, only using larger phrases when well motivated.[12]

This result alone is important: a model-based solution with no inference constraint has yielded a non-degenerate distribution over phrase lengths. Note that our sampler does find the degenerate solution quickly under a uniform prior, confirming that the model, and not the inference procedure, is selecting these small phrases.

4.6 A Hierarchical Dirichlet Process Prior

We also evaluate a hierarchical Dirichlet process (HDP) prior over θ_{J}, which draws monolingual distributions θ_{E} and θ_{F} from a DP and θ_{J} from their cross-product:

$$\begin{aligned} \theta_{\mathrm{J}} &\sim DP(M_0', \alpha) \\ M_0'(\langle e,f\rangle) &= [\theta_{\mathrm{F}}(f) P_{\mathrm{WA}}(e|f) \cdot \theta_{\mathrm{E}}(e) P_{\mathrm{WA}}(f|e)]^{\frac{1}{2}} \\ \theta_{\mathrm{F}} &\sim DP(P_f, \alpha') \\ \theta_{\mathrm{E}} &\sim DP(P_e, \alpha') \ . \end{aligned}$$

This prior encourages novel phrase pairs to be composed of phrases that have been used before. In the sampler, we approximate table counts for θ_{E} and θ_{F} with their expectations, which can be computed from phrase pair counts (see the appendix of Goldwater et al. (2006) for details). The HDP prior gives a similar distribution over phrase sizes.

5 Translation Results

We evaluate our new estimates using the baseline translation pipeline from the 2007 Statistical Machine Translation Workshop shared task.

5.1 Baseline System

We trained Moses on all Spanish-English Europarl sentences up to length 20 (177k sentences) using GIZA++ Model 4 word alignments and the *grow-diag-final-and* combination heuristic (Koehn et al., 2007; Och and Ney, 2003; Koehn, 2002), which performed better than any alternative combination heuristic.[13] The baseline estimates (*Heuristic*) come from extracting phrases up to length 7 from the word alignment. We used a bidirectional lexicalized distortion model that conditions on both foreign and English phrases, along with their orientations. Our 5-gram language model was trained on 38.3 million words of Europarl using Kneser-Ney smoothing. We report results with and without lexical weighting, denoted *lex*.

We tuned and tested on development corpora for the 2006 translation workshop. The parameters for each phrase table were tuned separately using minimum error rate training (Och, 2003). Results are

[11] Note that the expression for τ changes slightly under conditions where two phrase pairs being changed simultaneously coincidentally share the same lexical content. Details of these fringe conditions have been omitted for space, but were included in our implementation.

[12] The largest phrase pair found was 13 English words by 7 Spanish words.

[13] Sampling iteration time scales quadratically with sentence length. Short sentences were chosen to speed up our experiment cycle.

	Phrase		Exact
	Pair	NIST	Match
Estimate	**Count**	**BLEU**	**METEOR**
Heuristic	4.4M	29.8	52.4
DP	0.6M	28.8	51.7
HDP	0.3M	29.1	52.0
DP-composed	3.7M	30.1	52.7
HDP-composed	3.1M	30.1	52.6
DP-smooth	4.8M	30.1	52.5
HDP-smooth	4.6M	**30.2**	**52.7**
Heuristic + lex	4.4M	30.5	52.9
DP-smooth + lex	4.8M	30.4	53.0
HDP-smooth + lex	4.6M	**30.7**	**53.2**

Table 1: BLEU results for learned distributions improve over a heuristic baseline. Estimate labels are described fully in section 5.3. The label *lex* indicates the addition of a lexical weighting feature.

scored with lowercased, tokenized NIST BLEU, and exact match METEOR (Papineni et al., 2002; Lavie and Agarwal, 2007).

The baseline system gives a BLEU score of 29.8, which increases to 30.5 with *lex*, as shown in Table 1. For reference, training on all sentences of length less than 40 (the shared task baseline default) gives 32.4 BLEU with *lex*.

5.2 Learned Distribution Performance

We *initialized* the sampler with a configuration derived from the word alignments generated by the baseline. We greedily constructed a phrase alignment from the word alignment by identifying minimal phrase pairs consistent with the word alignment in each region of the sentence. We then ran the sampler for 100 iterations through the training data. Each iteration required 12 minutes under the DP prior, and 30 minutes under the HDP prior. Total running time for the HDP model neared two days on an eight-processor machine with 16 Gb of RAM.

Estimating phrase counts under the DP prior decreases BLEU to 28.8, or 29.1 under the HDP prior. This gap is not surprising: heuristic extraction discovers many more phrase pairs than sampling. Note that sacrificing only 0.7 BLEU while shrinking the phrase table by 92% is an appealing trade-off in resource-constrained settings.

5.3 Increasing Phrase Pair Coverage

The estimates *DP-composed* and *HDP-composed* in Table 1 take expectations of a more liberal count function. While sampling, we count not only aligned phrase pairs, but also larger ones composed of two or more contiguous aligned pairs. This count function is similar to the phrase pair extraction heuristic, but never includes unaligned phrases in any way. Expectations of these composite phrases still have a probabilistic interpretation, but they are not the structures we are directly modeling. Notably, these estimates outperform the baseline by 0.3 BLEU without ever extracting phrases from word alignments, and performance increases despite a reduction in table size.

We can instead increase coverage by smoothing the learned estimates with the heuristic counts. The estimates *DP-smooth* and *HDP-smooth* add counts extracted from word alignments to the sampler's running totals, which improves performance by 0.4 BLEU over the baseline. This smoothing balances the lower-bias sampler counts with the lower-variance heuristics ones.

6 Conclusion

Our novel Gibbs sampler and nonparametric priors together address two open problems in learning phrase alignment models, approximating inference consistently and efficiently while avoiding degenerate solutions. While improvements are modest relative to the highly developed word-alignment-centered baseline, we show for the first time competitive results from a system that uses word alignments only for model initialization and smoothing, rather than inference and estimation. We view this milestone as critical to eventually developing a clean probabilistic approach to machine translation that unifies model structure across both estimation and decoding, and decreases the use of heuristics.

References

David Aldous. 1985. Exchangeability and related topics. In *École d'été de probabilitiés de Saint-Flour*, Berlin. Springer.

Alexandra Birch, Chris Callison-Burch, and Miles Osborne. 2006. Constraining the phrase-based, joint probability statistical translation model. In *The Con-*

ference for the Association for Machine Translation in the Americas.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1994. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Colin Cherry and Dekang Lin. 2007. Inversion transduction grammar for joint phrasal translation modeling. In *The Annual Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Syntax and Structure in Statistical Translation*.

John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *The Annual Conference of the Association for Computational Linguistics: Short Paper Track*.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. 2006. Why generative phrase models underperform surface heuristics. In *The Annual Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Statistical Machine Translation*.

Thomas S Ferguson. 1973. A bayesian analysis of some nonparametric problems. In *Annals of Statistics*.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2007. The infinite tree. In *The Annual Conference of the Association for Computational Linguistics*.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *The Annual Conference of the Association for Computational Linguistics*.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric bayesian model. In *The Annual Conference of the Association for Computational Linguistics*.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *The Annual Conference of the Association for Computational Linguistics*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *The Annual Conference of the Association for Computational Linguistics*.

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

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *The Annual Conference of the Association for Computational Linguistics Workshop on Statistical Machine Translation*.

Daniel Marcu and Daniel Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *The Conference on Empirical Methods in Natural Language Processing*.

Jonathan May and Kevin Knight. 2007. Syntactic realignment models for machine translation. In *The Conference on Empirical Methods in Natural Language Processing*.

Robert Moore and Chris Quirk. 2007. An iteratively-trained segmentation-free phrase translation model for statistical machine translation. In *The Annual Conference of the Association for Computational Linguistics Workshop on Statistical Machine Translation*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *The Annual Conference of the Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *The Annual Conference of the Association for Computational Linguistics*.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *The Annual Conference of the Association for Computational Linguistics*.

Yee Whye Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *The Annual Conference of the Association for Computational Linguistics*.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *The Annual Conference of the Association for Computational Linguistics*.