Fast Collocation-Based Bayesian HMM Word Alignment

Philip Schulz
ILLC
University of Amsterdam
P.Schulz@uva.nl

Wilker Aziz
ILLC
University of Amsterdam
W.Aziz@uva.nl

Abstract

We present a new Bayesian HMM word alignment model for statistical machine translation. The model is a mixture of an alignment model and a language model. The alignment component is a Bayesian extension of the standard HMM. The language model component is responsible for the generation of words needed for source fluency reasons from source language context. This allows for untranslatable source words to remain unaligned and at the same time avoids the introduction of artificial NULL words which introduces unusually long alignment jumps. Existing Bayesian word alignment models are unpractically slow because they consider each target position when resampling a given alignment link. The sampling complexity therefore grows linearly in the target sentence length. In order to make our model useful in practice, we devise an auxiliary variable Gibbs sampler that allows us to resample alignment links in constant time independently of the target sentence length. This leads to considerable speed improvements. Experimental results show that our model performs as well as existing word alignment toolkits in terms of resulting BLEU score.

1 Introduction

Word alignment is one of the basic problems in statistical machine translation (SMT). The IBM models were originally devised for translation by Brown et al. (1993). Later, when SMT started to employ entire phrases instead of single words (Koehn et al., 2003), the IBM models were repurposed as word alignment models. The alignments they produce guide the phrase extraction heuristics that are used in many modern SMT systems.

There are several extensions of the classical IBM models that try to weaken their independence assumptions. Notably, Vogel et al. (1996) introduced Markovian dependencies between individual alignment links. Those links were treated as independent events in IBM models 1, 2 and 3. The model of Vogel et al. (1996) can be viewed as a Hidden Markov Model (HMM) in which the hidden Markov Chain induces a probability distribution over latent alignment links. Besides weakening the independence assumptions of the simpler IBM models, the HMM alignment model has the additional benefit of being tractable. This means that expectations under the HMM aligner can be computed exactly using the forward-backward algorithm. These expectations are then used in the Baum-Welch algorithm (Baum et al., 1970) to compute parameter updates for the model. Crucially, the Baum-Welch algorithm is a special case of the EM algorithm and thus guaranteed to never decrease the model's likelihood at each parameter update (Dempster et al., 1977).

Tractability and convergence (at least to a local optimum) are clear advantages of the HMM aligner over IBM models 3 to 5 which are all intractable. In practice, hill-climbing heuristics are employed to approximate expectations in these more complex models. Unfortunately, all convergence guarantees of the learning algorithm are lost this way.

A major problem for the HMM aligner is the handling of NULL words.[1] The NULL word is a special

[1]The original work of Vogel et al. (1996) did not use NULL words and instead aligned *all* source words. Later work by Och and Ney (2003) and Liang et al. (2006) has shown that using NULL words to allow for unaligned words improves performance.

lexical item that is hypothesised to stand at the beginning of every target (English) sentence. Since in word alignment the source (French) side is generated given the target side, the NULL word is used to generate source words that do not have lexical translations on the target side. Such untranslatable words are often idiosyncratic to the source language. Examples are prepositions such as *de* in French. The translation of the English *orange juice* is *jus **d'** orange*. In that case the (clitic) preposition *de* would be generated from the NULL word on the English side. For the HMM alignment model the NULL word is troublesome since it stands in the 0^{th} English position and thus induces unusually long jumps which have to be captured by the jump distribution of the HMM.

In this work we present a Bayesian HMM aligner that does not make use of artificial NULL words. Instead, untranslatable source words are generated from the words preceding them. This way, our proposed model only needs to account for lexically motivated alignment links. Moreover, since our model is a hierarchical Bayesian model we can bias it towards inducing sparse lexical distributions. This in turn leads to significantly better translation distributions (Mermer and Saraçlar, 2011). Moreover, we can even perform inference on the model's hyperparameters, freeing us of having to choose arbitrary prior distributions.

Existing Bayesian word aligners are often too slow to be useful in practice. We overcome this problem by designing an auxiliary variable Gibbs sampler that reduces sampling complexity by an order of magnitude. We also provide a formal proof that this sampler works correctly.

We provide several detailed experiments which show that our model performs on par with or better than standardly used alignment toolkits in terms of BLEU score.

Notation Throughout this paper we will denote random variables (RVs) by upper case Roman letters. If we want to express the probability of a specific outcome for the RV X we write $P(X = x)$. If we want to leave the value of X underspecified, we simply write $P(x)$. We abbreviate a sequence of RVs X_1 to X_n as X_1^n and a sequence of outcomes x_1 to x_n as x_1^n. We also distinguish notationally between probability mass functions (pmfs) and probability density functions (pdfs). We denote pmfs by $P(\cdot)$ and pdfs by $p(\cdot)$. Finally, we use v_F and v_E to denote the French (source) and English (target) vocabulary sizes.

2 The HMM Word Alignment Model

The HMM model of Vogel et al. (1996) defines a joint distribution over alignments and source words given a target sentence.[2] The source words are observed as part of the parallel corpus whereas the alignments are hidden. An alignment consists of as many alignment links as there are source words. Unlike the IBM models 1 and 2, which assume independence between alignment links, the HMM model assumes a first-order Markov dependency between them. This means that each alignment link depends on its predecessor.

We define random variables E ranging over the target vocabulary $\mathcal{E}$, variables F ranging over the source vocabulary $\mathcal{F}$, and variables A modelling alignment links. We use F_j for the variable associated with the source word occupying position j in the source sentence f_1^m. Consequently, A_j is the random variable for the alignment link associated with position j. This random variable ranges over word positions in the target sentence e_0^l, where 0 is a special position occupied by the hypothetical NULL word. Finally, we use e_{a_j} to denote the target word that position j is aligned to. We can then express the likelihood of the HMM model for a single sentence pair as shown in Equation 1.

$$P(f_1^m|e_0^l) = \prod_{j=1}^{m} \sum_{i=0}^{l} P(A_j = i|a_{j-1}) P(f_j|e_i) \tag{1}$$

Here, the Markovian dependency of the alignment links is captured by a distribution over *alignment jumps*. The distance between the current and the previous link can be thought of as a jump from one position to the next. The jump width is then given by $i - i'$ where i is the current value of A_j and i'

[2]The extension to a corpus of parallel sentences is trivial because independence between sentence pairs is assumed. To keep the notation simple we describe all models on the sentence level.

is the value of the previous alignment link A_{j-1}. The distribution over alignment jumps is then simply $P(A_j = i|A_{j-1} = i') = P(i - i')$. In practice, we set a_0 to a special start token, so as to provide a conditioning event for the first alignment decision.

A severe problem for the HMM alignment model is the handling of NULL words. If we did not treat alignments to the NULL position in a special way, those alignments would distort the alignment distribution because they induce unusually long jumps. To solve this problem, we introduce a special jump value for NULL alignments.[3] This has the effect that jumps to and from NULL are modelled explicitly and behave just like other jump values. On the other hand, if the preceding alignment is an alignment to NULL, the distribution over jumps becomes uniform. This is because when jumping from NULL a special jump value gets invoked that does not depend on the position that the next jump leads to. This behaviour obviously restricts the expressive power of the model somewhat but provides a clean handling of NULL alignments.

The parameters of the standard HMM of Vogel et al. (1996) are estimated through likelihood maximization. Here, we extend their model with prior distributions over parameters. This means that we are turning it into a Bayesian model. The parameters of the alignment HMM are categorical parameters of the following distributions:

- A distribution over the source lexicon for each target word. We call this the translation distribution and use Θ_e as a variable over the corresponding parameter vector.
- A distribution over alignment jumps. We call this the alignment distribution and use Θ_a as a variable over the corresponding parameter vector.

Since the model's distributions are categorical, we impose (symmetric) Dirichlet priors on their parameters. The Dirichlet is a standard choice in this case as it is conjugate to the categorical.[4] The variables in the Bayesian HMM aligner are thus generated as follows:

$$A_j|a_{j-1} \sim \theta_a \qquad \Theta_a \sim \mathrm{Dir}(\alpha)$$
$$F_j|e_{a_j} \sim \theta_{e_{a_j}} \qquad \Theta_e \sim \mathrm{Dir}(\beta)$$

We describe how to do inference in this model in Section 4.

3 An HMM Aligner with a Language Model Component

Our model is a mixture between a Bayesian HMM aligner and a Bayesian source language model. It differs from the Bayesian HMM aligner of Section 2 in that the language model component is the one responsible for generating source words from (source) context when those do not have a target translation. These are the words that would otherwise be aligned to NULL under the IBM models and our own Bayesian HMM (Section 2).

Formally, we extend the Bayesian HMM alignment model with binary choice variables Z which we are used to indicate collocations. There is one such variable for each source position, thus Z_j is the choice variable for position j. We use this variable as an indicator for whether the language model is used. Thus, if $Z_j = 1$, the source word f_j is generated from f_{j-1}. Otherwise, if $Z_j = 0$, f_j is generated from the target word it is aligned to (e_{a_j}).

Since our model is Bayesian, we put priors on all parameters. In particular, the translation parameters (θ_e), language model parameters (θ_f) and jump parameters (θ_a) are drawn from Dirichlet distributions with parameter vectors α, β and γ. The binary choice variables Z only have one parameter q_f which depends on the previous source word and follows a Beta distribution with parameters s and r. Since it is hard to guess reasonable values for the parameters s and r, we further assume that they are independently drawn from a Gamma distribution whose shape and rate parameters we set to 1. This extra level of hierarchy frees us from having to choose arbitrary values for s and r. Empirically, it also improves the performance of our model.

[3] Conceptually, we add a categorical event TONULL and another FROMNULL to the distribution over alignment jumps.

[4] Conjugacy in this case means that the posterior over parameters will again be a Dirichlet. This has the advantage that we can analytically integrate with respect to the posterior.

To better understand our model, we formulate a generative story:

- Draw values for s and r independently from $\mathrm{Gamma}(1, 1)$
- Generate q_f from $\mathrm{Beta}(s, r)$ for each source word f
- Draw values for θ_a, θ_e and θ_f from their respective Dirichlet priors
- For each source position j
  1. Generate an alignment link a_j conditional on a_{j-1}
  2. Choose a value for Z_j conditional on the previous source word f_{j-1}
     (a) If $Z_j = 0$, generate source word f_j from target word e_{a_j}
     (b) If $Z_j = 1$, generate source word f_j from source word f_{j-1}

In terms of variable generation we have to adjust the model description from Section 2. For reasons of clarity, we condition all variables only on those events that they depend on.

$$
\begin{aligned}
S &\sim \mathrm{Gamma}(1,1) & R &\sim \mathrm{Gamma}(1,1) \\
Z_j|f_{j-1} &\sim \mathrm{Bernoulli}(q_{f_{j-1}}) & Q_f &\sim \mathrm{Beta}(s,r) \\
A_j|a_{j-1} &\sim \mathrm{Cat}(\theta_a) & \Theta_a &\sim \mathrm{Dir}(\alpha) \\
F_j|a_j, Z_j = 0, e_{a_j} &\sim \mathrm{Cat}(\theta_{e_{a_j}}) & \Theta_e &\sim \mathrm{Dir}(\beta) \\
F_j|a_j, Z_j = 1, f_{j-1} &\sim \mathrm{Cat}(\theta_{f_{j-1}}) & \Theta_f &\sim \mathrm{Dir}(\gamma)
\end{aligned}
$$

Our model as described above defines a joint distribution over French words, collocation and alignment variables and parameters given an English sentence and the hyperparameters.

4 Inference

In this section we derive a Gibbs sampler for our collocation-based model (Section 3). The sampler for the Bayesian HMM with NULL words (Section 2) follows from that. We also describe how to sample the hyperparameters of the Beta prior on the collocation distributions.

4.1 Dirichlet Predictive Posterior

Let us first establish a general useful fact about the Dirichlet distribution.[5] Let us call the posterior Dirichlet parameter vector η.[6] Then the posterior predictive distribution for a categorical outcome x is given by the following integral:

$$P(x|\eta) = \int P(x|\theta)p(\theta|\eta)d\theta = \int \theta_x p(\theta|\eta)d\theta \; . \tag{2}$$

Here, we use θ_x to denote the categorical parameter for outcome x. Note that the last integral in Equation (2) is in fact equal to the expectation of θ_x under $\mathrm{Dir}(\eta)$. It is well known that for a k-dimensional Dirichlet distribution this expectation is $\mathbb{E}[\theta_x|\eta] = \frac{\eta_x}{\sum_{i=1}^{i=k} \eta_i}$. Thus, the predictive posteriors for the alignment and collocation variables after integrating over the categorical or Beta parameters take the form of this expectation.

4.2 Gibbs Sampling the Hidden Variables

Since we are only interested in the assignments of the alignment and collocation variables, we integrate over the model parameters θ_e, θ_a, θ_f and q_f. This gives us a collapsed Gibbs sampler.

In general, a Gibbs sampler resamples one variable at a time while conditioning on the current assignments of all other variables. In our case we are interested in resampling A_j and Z_j. This means that we

[5] This fact immediately carries over to the Beta distribution which can be viewed as a 2-dimensional Dirichlet distribution.

[6] Since the categorical is in the exponential family and the Dirichlet is conjugate to it, the posterior Dirichlet parameter η is simply the sum of the prior Dirichlet parameter (which we will call ϵ here) and the sufficient statistics of the categorical likelihood. In our concrete case, these sufficient statistics are simply the number of times an outcome has been observed in the data. Let us denote these counts by $c(x)$ for an outcome x. Thus, for this outcome x, we have $\eta_x = \epsilon_x + c(x)$.

will need to derive posterior predictive distributions for these variables. Let us first deal with the posterior predictive for the collocation variables where we assume the Beta parameters r and s as given. To ease notation we introduce the set $\mathcal{C} = \{r, s, \alpha, \beta, \gamma\}$ which contains all remaining (hyper-)parameters. We use $x_{\neg j}$ to denote the set of all outcomes x except the j^{th} one. We also introduce the function $c(\cdot)$ as a (conditional) count function for an outcome across the entire corpus, excluding the j^{th} such outcome if necessary. Proportionality in the following equations comes about because we eliminate normalization constants that do not depend on the value of the variable that we sample.

$$\begin{aligned} P(Z_j = 0 | z_{\neg j}, a_1^m, f_{\neg j}, e_1^l, \mathcal{C}) &\propto P(Z_j = 0 | f_{j-1}, s, r) \times P(f_j | a_j, e_{a_j}, \beta) \\ &= \frac{c(z = 0 | f_{j-1}) + r}{c(f_{j-1}) + r + s} \times \frac{c(f_j | e_{a_j}) + \beta}{c(e_{a_j}) + v_F \beta} \\ &\propto (c(z = 0 | f_{j-1}) + r) \times \frac{c(f_j | e_{a_j}) + \beta}{c(e_{a_j}) + v_F \beta} \end{aligned} \tag{3}$$

$$\begin{aligned} P(Z_j = 1 | z_{\neg j}, a_1^m, f_{\neg j}, e_1^l, \mathcal{C}) &\propto P(Z_j = 1 | f_{j-1}, s, r) \times P(f_j | f_{j-1}, \gamma) \\ &= \frac{c(z = 1 | f_{j-1}) + s}{c(f_{j-1}) + r + s} \times \frac{c(f_j | f_{j-1}) + \gamma}{c(z = 1 | f_{j-1}) + v_F \gamma} \\ &\propto (c(z = 1 | f_{j-1}) + s) \times \frac{c(f_j | f_{j-1}) + \gamma}{c(z = 1 | f_{j-1}) + v_F \gamma} \end{aligned} \tag{4}$$

When resampling the alignment variables, we need to distinguish between the two cases where the collocation variable is active or not. In case the collocation variable is switched off, inference for the alignment variable is similar to that in the Bayesian HMM aligner. The crucial difference, however, is that in the standard case, the values of the alignment variable range from 0, the NULL position, to the target sentence length l. In our model, however, the possible values for the alignment variable start at 1 as there is no NULL position.

$$\begin{aligned} P(a_j | Z_j = 0, a_{\neg j}, f_1^m, e_1^l, \mathcal{C}) &\propto P(a_j | a_{j-1}, \alpha) \times P(a_{j+1} | a_j, \alpha) \times P(f_j | e_{a_j}, \beta) \\ = \frac{c(a_j - a_{j-1}) + \alpha}{\sum_{i=1}^{l} (c(i - a_{j-1}) + \alpha)} &\times \frac{c(a_{j+1} - a_j) + \alpha}{\sum_{i=1}^{l} (c(a_{j+1} - i) + \alpha)} \times \frac{c(f_j | e_{a_j}) + \beta}{c(e_{a_j}) + v_F \beta} \\ &\propto (c(a_j - a_{j-1}) + \alpha) \times (c(a_{j+1} - a_j) + \alpha) \times \frac{c(f_j | e_{a_j}) + \beta}{c(e_{a_j}) + v_F \beta} \end{aligned} \tag{5}$$

Again, let us point out that the predictive posterior in (5) is the one we use for inference in the Bayesian HMM model described in Section 2 with difference that the alignment positions include 0 in that case.

If the collocation variable is switched on, i.e. if the j^{th} French word is generated from its predecessor, there is no lexical influence on the alignment posterior and the new link is simply sampled from the alignment distribution.

$$\begin{aligned} P(a_j | Z_j = 1, a_{\neg j}, f_{\neg j}, e_1^l, \mathcal{C}) &\propto P(a_j | a_{j-1}, \alpha) \times P(a_{j+1} | a_j, \alpha) \\ = \frac{c(a_j - a_{j-1}) + \alpha}{\sum_{i=1}^{l} (c(i - a_{j-1}) + \alpha)} &\times \frac{c(a_{j+1} - a_j) + \alpha}{\sum_{i=1}^{l} (c(a_{j+1} - i) + \alpha)} \\ &\propto (c(a_j - a_{j-1}) + \alpha) \times (c(a_{j+1} - a_j) + \alpha) \end{aligned} \tag{6}$$

At this point it is important to note that the naïve Gibbs sampler described here considers all target positions as candidate alignment points for a given source position j. This means that the time it takes to re-sample an alignment link grows linearly with the length of the target sentence. In practice this makes the sampler unpractically slow. We address this problem in the following section.

4.3 Sampling Alignments Efficiently

We are interested in sampling from $P(a_j|a_{\neg j}, \mathcal{C})$, where we conflate all other variables in $\mathcal{C}$ to avoid clutter. Note that we know $P(a_j|a_{\neg j}, \mathcal{C})$ up to a normalisation constant, thus sampling requires evaluating Equation 5 (or 6) for all $1 \leq i \leq l$. Naturally, this procedure runs in time proportional to $O(l)$. In this section we present an auxiliary variable sampler that brings this down to constant time.

The idea is to evaluate $P(a_j|a_{\neg j}, \mathcal{C})$ only for assignments in a subset of target positions.[7] This subset must include at least 2 candidates and it must be such that it contains the current assignment of the variable we are resampling. That is, if we let $\left(a_{\neg j}^{(t)}, a_j^{(t)}\right)$ denote the current state of the Markov chain, then we need to select $a_j^{(t)}$ as well as at least one random candidate from the remaining available positions $\{1, \ldots, l\} \setminus \{a_j^{(t)}\}$. We denote a selection of target positions by a vector $k \subseteq \{0,1\}^l$ such that $k_i = 1$ signifies that i is a reachable candidate. Then, once a selection is made, we sample the next state of the Markov chain, i.e. $\left(a_{\neg j}^{(t)}, a_j^{(t+1)}\right)$, from a distribution proportional to $P(A_j^{(t+1)} = i|a_{\neg j}^{(t)}, \mathcal{C}^{(t)}) \times k_i$. Note that, under this distribution, only selected positions have non-zero probability. This makes sampling the next state run in constant time independent of the target sentence length.

Formally, this is a case of sampling by data augmentation (Tanner and Wong, 1987). Let K be a random variable taking values in $\mathcal{K} \subset \{0,1\}^l$. We interpret an assignment of K as a random selection of at most l target positions. We define the joint distribution $P(a_j, k|a_{\neg j}, \mathcal{C}) = P(a_j|a_{\neg j}, \mathcal{C}) \times P(k|a_j, a_{\neg j})$, where we take the conditional $P(k|A_j = i, a_{\neg j}) = \frac{k_i}{\sum_{k' \in \mathcal{K}} k'_i}$ to distribute uniformly over all selections in $\mathcal{K}$ that contain i. This guarantees that the current state of the Markov chain is part of the selection, a condition necessary for irreducibility. Then, the conditional $P(A_j = i'|k, a_{\neg j}, \mathcal{C})$ follows directly:

$$P(A_j = i'|k, a_{\neg j}, \mathcal{C}) \propto P(A_j = i'|a_{\neg j}, \mathcal{C}) \times P(k|A_j = i', a_{\neg j}) \propto P(A_j = i'|a_{\neg j}, \mathcal{C}) \times k_{i'} \quad (7)$$

Claim If $\mathcal{K}$ includes at least all subsets of size 2 where one of the elements is the previous state of the Markov chain, then the transition kernel $\kappa(i'|i) = \sum_{k \in \mathcal{K}} P(i'|k) \times P(k|i)$ is strictly positive for every $i, i' \in \{1, \ldots, l\}$ and, therefore, the resulting Markov chain is Harris ergodic.

Proof.

$$\kappa(i'|i) = \sum_{k \in \mathcal{K}} P(i'|k) \times P(k|i) = \sum_{k \in \mathcal{K}} \frac{P(i') \times P(k|i')}{P(k)} \times P(k|i) = P(i') \sum_{k \in \mathcal{K}} \frac{\frac{k_{i'}}{\sum_{k' \in \mathcal{K}} k'_{i'}} \times \frac{k_i}{\sum_{k' \in \mathcal{K}} k'_i}}{P(k)} \quad (8)$$

In the last term of Equation 8, $P(i')$ is strictly positive because $i' \in \{1, \ldots, l\}$ by construction, and the sum is strictly positive when $\mathcal{K}$ includes at least one subset containing both i and i'. This is why we can start the candidate set with $\{a_j^{(t)}\}$ and enlarge it by sampling uniformly from $\{1, \ldots, l\} \setminus \{a_j^{(t)}\}$ without replacement. We need to do it at least once, but we can also repeat it a fixed number of times. □

The complete algorithm consists of 2 simulations, $K^{(t+1)} \sim P(\cdot|a_j^{(t)}, a_{\neg j}^{(t)})$ and $A_j^{(t+1)} \sim P(\cdot|k^{(t+1)}, a_{\neg j}^{(t)}, \mathcal{C}^{(t)})$, each feasible in isolation. With this improved sampler we can resample alignment links in constant time whereas the naïve sampler would require time linear in target sentence length. It is easy to see that to resample all alignment links in a source sentence with m words, the naïve sampler would take time $O(l \times m) \approx O(l^2)$, whereas our improved auxiliary variable sampler does the same in $O(l)$. This improvement makes our model competitive with maximum-likelihood models.

4.4 Sampling of the Beta Parameters

The parameters s and r of the Beta distribution which serves as a prior on the decision variables are random variables in our model. Since there is no easily computable conjugate distribution for these parameters, we can not integrate them out analytically. Instead, we choose to approximate the integral

[7] Recall that l is the target sentence length.

through repeated sampling of these variables. Since both these variables take on values in the positive reals, we impose a Gamma prior on them as described in Section 3.

There are several ways of sampling variables in non-conjugate Bayesian models. Here we use slice sampling (Neal, 2003) as it is fast and easy to implement. The idea of slice sampling is that we augment our sampling distribution with an auxiliary variable U such that the marginal distribution of S (or R) stays unchanged.[8] Simulation then follows by Gibbs sampling, whereby we sample U conditioned on S, and S conditioned on U in turn. It can be shown that, with conditionals as shown in (9), the transition kernel underlying this Gibbs sampling procedure is Harris ergodic (Neal, 2003). Thus, the procedure is not only correct but also very efficient as we only sample from uniform distributions.

$$p(u|s) = \frac{\mathbb{1}(u < p(s))}{p(s)} \qquad p(s|u) \propto \begin{cases} 1 & \text{if } p(s) \geq u \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The posterior that we slice sample from is simply proportional to the product of likelihood and prior of the Beta distribution: $p(s|f_1^m, z_1^m, r) \propto P(z_1^m|f_1^m, s, r) \times p(s)$.

Notice that the conditions in Equation (9) guarantee that at least the current point will be in the slice. We will therefore always be able to obtain a new sample. Intuitively, slice sampling works because the marginal distribution $p(s)$ stays unchanged.

4.5 Decoding

After we have taken a number of samples, we are ready to decode. We use a version of maximum marginal decoding (Johnson and Goldwater, 2009) in which we assign to each source position j the target position that was most often sampled as a value for A_j. If most of the time the collocation variable Z_j was active, however, we leave that source position unaligned.

5 Experiments and Results

All experiments were run using the Moses phrase-based system (Koehn et al., 2007) with lexicalized reordering. In order to speed up our experiments we used cube pruning with a pop limit of 1000 in both tuning and evaluation. Symmetrised alignments were obtained with the `grow-diag-final-and` heuristic.

Data We used the WMT 2014 news commentary data[9] to train our models and the corresponding dev (`newstest2013`) and test (`newstest2014`) sets for tuning and evaluation. We use all available monolingual data to train 5-gram language models with KenLM (Heafield, 2011).

Models We report results for the Bayesian HMM described in Section 2 (BHMM) and our collocation-based model described in Section 3 (BHMM-Z). To enable comparison with standard alignment toolkits, we also report results with Giza++ and fastAlign (Dyer et al., 2013). Finally, we make a comparison with the collocation-based IBM2 model of Schulz et al. (2016) (BIBM2-Z).

Hyperparameters The hyperparameters of our model were set to $\beta = \gamma = 0.0001$ to obtain sparse lexical distributions. For the jump prior we chose $\alpha = 1$, not giving preference to any particular distribution. The same choices apply for the BIBM2-Z. However, that model does not employ hyperparameter inference and we therefore set $s = 1, r = 0.01$. Finally, the BHMM uses the same parameters as the previous two models, except those associated with the collocation variable.

All samplers were run for 1000 iterations without burn-in and samples were taken after each 25^{th} iteration. The initial state was chosen to be the Viterbi decoding of IBM model 1. For our auxiliary variable sampler we select exactly 2 candidates, thus making inference as fast as possible. Giza++ and fastAlign were run under their standard parameter settings. For Giza++ this means that we ran EM for IBM model 1 and the HMM (5 iterations each) and for IBM models 3 and 4 (3 iterations each). Since

[8] In the following exposition we will describe the resampling of S and assume a fixed value r. Resampling R works analogously with a fixed value s.

[9] `http://statmt.org/wmt14/translation-task.html`

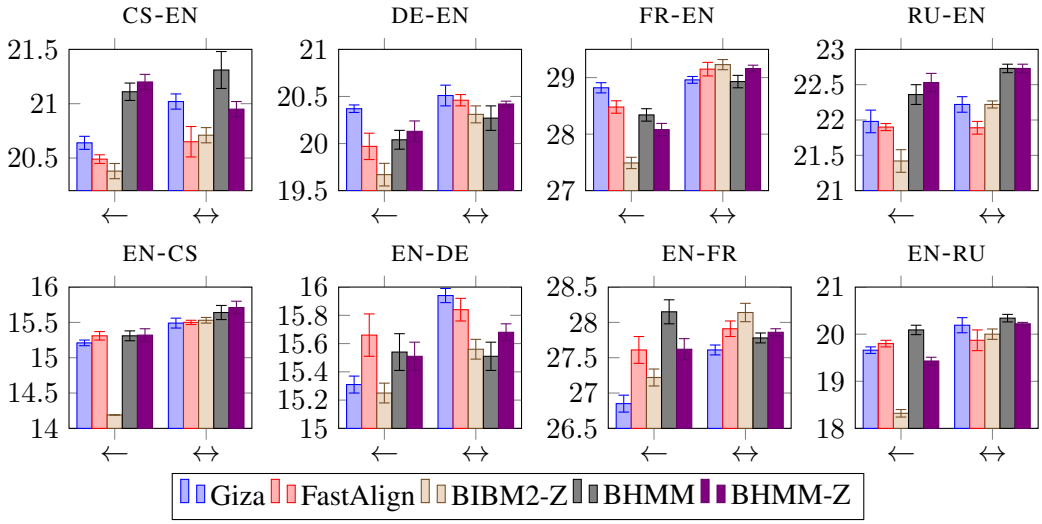

Figure 1: BLEU scores for each translation direction trained on (←) directional (condition on target and generate source) and (↔) symmetrised alignments (`grow-diag-final-and`). Observe that the plots are on different scales. This means that results cannot directly be compared across plots.

the maximum likelihood HMM is incorporated in the Giza++ pipeline, we do not report separate results for it.

Results Figure 1 shows translation quality results in terms of BLEU for different aligners, where BHMM is the Bayesian HMM (Section 2) and BHMM-Z is our novel collocation-based model (Section 3). Furthermore, BIBM2-Z is the model of Schulz et al. (2016). To account for optimiser instability (MERT in this case), we plot average and standard deviation across 5 independent runs. Note that our Bayesian models perform mostly on par with maximum-likelihood models. In the directional case, our Bayesian models lose to Giza++ only in DE-EN by a very slim margin. Except for EN-DE (symmetrised) our models are never worse than fastAlign. Moreover, our models perform notably well on Czech and Russian outperforming the maximum-likelihood models. Finally, except for EN-FR (directional), our collocation-based BHMM-Z improves upon the more basic BHMM. It also often improves upon the BIBM2-Z. That model is only better on the symmetrised English-French data where it outperforms all other models.

We also analysed the number of alignment links that our systems set. It is noteworthy that the Bayesian HMM with NULL words consistently sets much fewer links than all other systems. Taking BHMM as baseline other models set additionally (on average across languages and translation direction) 39.5% (our collocation-based model), 39.2% (Giza++), and 36.2% (fastAlign) more links. Setting more links constrains phrase extraction heuristics more and leads to smaller phrase tables. Empirically, the phrase tables of the collocation-based model are roughly three times smaller than those of the Bayesian HMM. Thus, the collocation-based system is at an advantage here.

Finally, in terms of speed, Giza++ takes on average (across languages and translation directions) 202 minutes on 2 CPUs, while BHMM and BHMM-Z take respectively 81 and 267 minutes in 1 CPU.[10] The slower performance of BHMM-Z in relation to BHMM is not per se due to the sampling of collocation variables, but rather due to hyperparameter inference (see Section 4.4).

6 Related Work

There are several extensions of the classical IBM models. Dyer et al. (2013) use a log-linear model for the distortion distribution. While they keep the independence assumptions for alignment links, they bias

[10]The run times for our models include the computation of the initial state of the sampler with IBM model 1. The run time of the sampler itself is thus slightly lower than the reported times. Also notice that due to its highly optimised posterior computations, fastAlign finishes in under 10 minutes on average.

their model to preferably align positions which are close to each other. Using standard results for series, they manage to make the posterior computations in their model extremely fast.

Another interesting extension of the HMM alignment is presented in Zhao and Gildea (2010) who added a fertility distribution to the HMM. This made posterior computations in their model intractable, however, they avoided the use of heuristics and instead approximated the posterior using MCMC.

The idea of using Bayesian inference together with a Gibbs sampler for word alignment was first presented for IBM model 1 by Mermer and Saraçlar (2011). They also gave a more detailed analysis of their method and extended it to IBM model 2 in Mermer et al. (2013). The model presented here is also similar in spirit to Schulz et al. (2016) who proposed to use a language model component together with an alignment model. However, they used IBM models 1 and 2 as alignment components.

Apart from the present work, the only other work on Bayesian word alignment that we know of that performed as well as or better than Giza++ was presented in Gal and Blunsom (2013). These authors reformulated IBM models 1-4 as hierarchical Pitman-Yor processes. While their models are highly expressive, the Gibbs sampler based on Chinese Restaurant processes that they used is very slow and thus their models are unfortunately not useful in practice.

7 Discussion and Future Work

We envision several useful extensions of our model for the future. Firstly, we plan to turn the language model distribution into a hierarchical distribution. We plan to use either a hierarchical Dirichlet process or Pitman-Yor process for this. The advantage of this technique is that information about untranslatable source words can be shared across preceding source words. Sampling from such a distribution using a Chinese Restaurant Process is potentially time-consuming. However, we are confident that we can maintain a good speed if we apply our auxiliary variable technique and employ efficient samplers as described by Blei and Frazier (2011).

Since our model aligns many words and thus restricts the amount of phrases that can be extracted, it may also be a good alignment model for hierarchical phrase-based SMT. We plan to apply our model to this scenario in the future.

The code used in our experiments is freely available at `https://github.com/philschulz/Aligner`.

Acknowledgements

We would like to thank Stella Frank for helpful discussions about hyperparameter inference. We also thank our reviewers for their excellent feedback. This work was supported by the Dutch Organization for Scientific Research (NWO) VICI Grant nr. 277-89-002.

References

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

David M. Blei and Peter I. Frazier. 2011. Distance dependent Chinese Restaurant Processes. *Journal of Machine Learning Research*, 12:2461–2488.

Peter F. Brown, Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

Yarin Gal and Phil Blunsom. 2013. A systematic bayesian treatment of the IBM alignment models. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 969–977.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325, Boulder, Colorado, June. Association for Computational Linguistics.

Philipp Koehn, 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-Volume 1*, pages 48–54. 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, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180.

Percy Liang, 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 City, USA, June. Association for Computational Linguistics.

Coşkun Mermer and Murat Saraçlar. 2011. Bayesian word alignment for statistical machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, pages 182–187.

Coşkun Mermer, Murat Saraçlar, and Ruhi Sarikaya. 2013. Improving statistical machine translation using Bayesian word alignment and Gibbs sampling. *IEEE Transactions on Audio, Speech & Language Processing*, 21(5):1090–1101.

Radford M. Neal. 2003. Slice sampling. *Annals of Statistics*, 31:705–767.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Philip Schulz, Wilker Aziz, and Sima'an Khalil. 2016. Word alignment without NULL words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Martin A. Tanner and Wing Hung Wong. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, June.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, pages 836–841, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shaojun Zhao and Daniel Gildea. 2010. A fast fertility hidden Markov model for word alignment using MCMC. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 596–605, Cambridge, MA, October. Association for Computational Linguistics.