PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 106 OCTOBER 2016 125–146

Efficient Word Alignment with Markov Chain Monte Carlo

Robert Östling, Jörg Tiedemann

Department of Modern Languages, University of Helsinki

Abstract

We present EFMARAL, a new system for efficient and accurate word alignment using a Bayesian model with Markov Chain Monte Carlo (MCMC) inference. Through careful selection of data structures and model architecture we are able to surpass the FAST_ALIGN system, commonly used for performance-critical word alignment, both in computational efficiency and alignment accuracy. Our evaluation shows that a phrase-based statistical machine translation (SMT) system produces translations of higher quality when using word alignments from EFMARAL than from FAST_ALIGN, and that translation quality is on par with what is obtained using GIZA++, a tool requiring orders of magnitude more processing time. More generally we hope to convince the reader that Monte Carlo sampling, rather than being viewed as a slow method of last resort, should actually be the method of choice for the SMT practitioner and others interested in word alignment.

1. Introduction

Word alignment is an essential step in several applications, perhaps most prominently phrase-based statistical machine translation (Koehn et al., 2003) and annotation transfer (e.g. Yarowsky et al., 2001). The problem is this: given a pair of translationally equivalent sentences, identify which word(s) in one language corresponds to which word(s) in the other language. A number of off-the-shelf tools exist to solve this problem, but they tend to be slow, inaccurate, or both. We introduce EFMARAL, a new open-source tool[1] for word alignment based on partially collapsed Gibbs sampling in a Bayesian model.

[1]The source code and documentation can be found at `https://github.com/robertostling/efmaral`

 Corresponding author: `robert.ostling@helsinki.fi`
Cite as: Robert Östling, Jörg Tiedemann. Efficient Word Alignment with Markov Chain Monte Carlo. The Prague Bulletin of Mathematical Linguistics No. 106, 2016, pp. 125–146. doi: 10.1515/pralin-2016-0013.

2. Background

In order to understand the present work, we first need to formalize the problem and introduce the family of models used (Section 2.1), describe their Bayesian extension (Section 2.2), the Markov Chain Monte Carlo algorithm used for inference (Section 2.3) and its particular application to our problem (Section 2.4).

2.1. The IBM models

The IBM models (Brown et al., 1993) are asymmetric generative models that describe how a *source language* sentence generates a *target language* sentence though a set of latent alignment variables. Since the task at hand is to align the words in the source and target language sentences, the words in both sentences are given, and we are left with inferring the values of the alignment variables.

Formally, we denote the k:th sentence pair $\langle \mathbf{s}^{(k)}, \mathbf{t}^{(k)} \rangle$ with the source sentence $\mathbf{s}^{(k)}$ containing words $s_i^{(k)}$ (for each word index $i \in 1 \dots I^{(k)}$) and the target sentence $\mathbf{t}^{(k)}$ containing words $t_j^{(k)}$ (for $j \in 1 \dots J^{(k)}$).

Each sentence pair $\langle \mathbf{s}^{(k)}, \mathbf{t}^{(k)} \rangle$ is associated with an alignment variable $\mathbf{a}^{(k)}$, where $a_j^{(k)} = i$ indicates that target word $t_j^{(k)}$ was generated by source word $s_i^{(k)}$. This implies an n-to-1 mapping between source and target words, since each target word is aligned to exactly one source word, while each source word can be aligned to zero or more target words.

Sentences are assumed to be generated independently, so the probability of generating a set of parallel sentences $\langle \mathbf{s}, \mathbf{t} \rangle$ is

$$P(\mathbf{t}|\mathbf{s}, \mathbf{a}) = \prod_{k=1}^{K} P(\mathbf{t}^{(k)}|\mathbf{s}^{(k)}, \mathbf{a}^{(k)}) \tag{1}$$

For simplicity of notation, we will drop the sentence index (k) in the following discussion and let $\langle \mathbf{s}, \mathbf{t} \rangle$ instead denote a single sentence pair, without loss of generality due to the independence assumption between sentences.

A source word type e is associated with a *lexical distribution*, modeled by a categorical distribution with parameter vector θ_e. In the simplest of the IBM models (model 1), the probability of generating a target sentence $\mathbf{t}$ is defined as the probability of independently generating each of the J target words independently from the lexical distributions of their respective aligned source words.

$$P(\mathbf{t}|\mathbf{s}, \mathbf{a}) \propto \prod_{j=1}^{J} \theta_{s_{a_j}, t_j} \tag{2}$$

IBM model 1 assumes a uniform distribution for $P(\mathbf{a})$, which effectively means that the word order of the sentences are considered irrelevant. This is clearly not true

in real translated sentences, and in fact a_j and a_{j+1} tend to be strongly correlated. Most research on word alignment has assumed some version of a *word order model* to capture this dependency. Perhaps the simplest version is used in IBM model 2 and the FAST_ALIGN model (Dyer et al., 2013), which are based on the observation that $j/J \approx a_j/I$, in other words that sentences tend to have the same order of words in both languages. This is however a very rough approximation, and Vogel et al. (1996) instead proposed to directly model $P(a_{j+1} - a_j = x|I)$, which describes the length x of the "jump" in the source sentence when moving one word forward in the target sentence, conditioned on the source sentence length I.

Although the IBM models allow n-to-1 alignments, not all values of n are equally likely. In general, high values of n are unlikely, and a large proportion of translations are in fact 1-to-1. The value of n depends both on the particular languages involved (a highly synthetic language like Finnish translated into English would yield higher values than a French to English translation) and on the specific word type. For instance, the German *Katze* 'cat' would typically be translated into a single English word, whereas *Unabhängigkeitserklärung* would normally be translated into two (*independence declaration*) or three words (*declaration of independence*). This can be modeled by defining the *fertility* $\phi(i) = \sum_{j=1}^{J} \delta_{a_j=i}$ of a source token s_i, and introducing a distribution for $P(\phi(i) = n|s_i = e)$ for each source word type e.

A large number of models based on the same general assumptions have been explored (Brown et al., 1993; Toutanova et al., 2002; Och and Ney, 2003), and the interested reader may want to consult Tiedemann (2011) for a more thorough review than we are able to provide in this work.

2.2. Bayesian IBM models

The IBM models make no a priori assumptions about the categorical distributions that define the model, and most authors have used maximum-likelihood estimation through the Expectation-Maximization algorithm (Dempster et al., 1977) or some approximation to it. However, when translating natural languages the lexical distributions should be very sparse, reflecting the fact that a given source word tends to have a rather small number of target words as allowable translations, while the vast majority of target words are unimaginable as translations.

These constraints have recently been modeled with sparse and symmetric Dirichlet priors (Mermer and Saraçlar, 2011; Mermer et al., 2013; Riley and Gildea, 2012) which, beyond capturing the range of lexical distributions we consider likely, also turn out to be mathematically very convenient as the Dirichlet distribution is a conjugate prior to the categorical distribution. The d-dimensional Dirichlet distribution is defined over the space of d-dimensional categorical distributions, and is parameterized by the d-dimensional vector $\boldsymbol{\alpha} > 0$. If $\mathbf{X} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, the probability density function of $\mathbf{X}$ is

given by

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{d} x_i^{\alpha_i - 1} \tag{3}$$

where the normalization constant Z is given by the multinomial beta function

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{d} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{d} \alpha_i)} \tag{4}$$

A symmetric Dirichlet distribution has $\alpha_i = \alpha_j$ for all i, j, with the interpretation in our case that no particular translation is preferred a priori for any source word type, as this has to be estimated from the data. By also setting $\alpha \ll 1$ we favor sparse lexical distributions where most probabilities are close to zero.

While it is possible to treat $\boldsymbol{\alpha}$ as a latent variable to be inferred, good results can be obtained by using a fixed value roughly in the range of 10^{-6} to 10^{-2} (Riley and Gildea, 2012). Another direction of research has explored hierarchical distributions such as the Pitman-Yor process (Pitman and Yor, 1997) instead of the Dirichlet distribution for the translation distribution priors (Gal and Blunsom, 2013; Östling, 2015). Such distributions offer even greater flexibility in specifying prior constraints on the categorical distributions, but at the cost of less efficient inference. Since the gain in accuracy has turned out to be limited and computational efficiency is an important concern to us, we will not further consider hierarchical priors in this work.

2.3. Markov Chain Monte Carlo

Several different methods have been used for inference in IBM alignment models. Starting with Brown et al. (1993), maximum-likelihood estimation through the Expectation-Maximization (EM) algorithm has been a popular choice. This method is generally efficient for simple models without word order or fertility distributions, but computing the expectations becomes intractable for more complex models such as IBM model 4 so approximative hill-climbing methods are used instead.

Another disadvantage of using plain EM inference with the IBM models is that it is unable to incorporate priors on the model parameters, and as was pointed out in the previous section this deprives us of a powerful tool to steer the model towards more realistic solutions. Riley and Gildea (2012) presented a method to extend the EM algorithm to IBM models with Dirichlet priors, through Variational Bayes inference. Unfortunately, their method inherits the complexity issues of earlier EM approaches.

The inference approach chosen by most authors working on Bayesian IBM models (Mermer and Saraçlar, 2011; Gal and Blunsom, 2013; Östling, 2015) is Gibbs sampling (Gelfand and Smith, 1991), a special case of the Markov Chain Monte Carlo (MCMC) method which we will briefly summarize here.

Given a probability function $p_M(\mathbf{x})$ of some model M on parameter vector $\mathbf{x}$, MCMC provides us with the means to draw samples from p_M. This is done by constructing a Markov chain with values of $\mathbf{x}$ as states, such that its stationary distribution is identical to p_M. In practice, this means deriving expressions for the transition probabilities $P(\mathbf{x}'|\mathbf{x})$ of going from state $\mathbf{x}$ to state $\mathbf{x}'$. Since the number of states is enormous or infinite in typical applications, it is essential that there is some way of sampling efficiently from $P(\mathbf{x}'|\mathbf{x})$. With Gibbs sampling, this is done by sampling one variable from the parameter vector $\mathbf{x}$ at a time, conditioned on all other variables: $P(x_i|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$ which we will write as $P(x_i|\mathbf{x}^{(-i)})$ to indicate conditioning on all elements of x except at index i. All positions i are then sampled in some arbitrary but fixed order. By choosing suitable distributions for the model, the goal in designing a Gibbs sampler is to make sure that this distribution is easy to sample from.

2.4. Gibbs sampling for Bayesian IBM models

The Bayesian version of IBM model 1 defines the following probability over the parameter vector, which consists of the alignment vector $\mathbf{a}$ and the lexical distribution vectors θ_e for each e in the source target vocabulary:

$$P(\mathbf{a}, \boldsymbol{\theta}) = P(\mathbf{s}, \mathbf{t}, \mathbf{a}, \boldsymbol{\theta}, \boldsymbol{\alpha}) \propto \left(\prod_{k=1}^{K} \prod_{j=1}^{J^{(k)}} \theta_{s^{(k)}_{a^{(k)}_j}, t^{(k)}_j} \right) \cdot \left(\prod_{e=1}^{E} \prod_{f=1}^{F} \theta_{e,f}^{\alpha_f - 1} \right) \tag{5}$$

since $\mathbf{s}$, $\mathbf{t}$ and $\boldsymbol{\alpha}$ are constant.

A straightforward Gibbs sampler can be derived by observing that

$$P(x_i|\mathbf{x}^{(-i)}) = \frac{P(\mathbf{x})}{P(\mathbf{x}^{(-i)})} = \frac{P(\mathbf{x}^{(-i)}, x_i)}{P(\mathbf{x}^{(-i)})}$$

which means that

$$P(a_j = i|\mathbf{a}^{(-j)}, \boldsymbol{\theta}) = \frac{P(\mathbf{a}^{(-j)}, a_j = i, \boldsymbol{\theta})}{P(\mathbf{a}^{(-j)}, \boldsymbol{\theta})} \propto \theta_{s_{a_j}, t_j} \tag{6}$$

and

$$P(\boldsymbol{\theta}_e = x|\mathbf{a}, \boldsymbol{\theta}^{(-e)}) = \frac{P(\boldsymbol{\theta}^{(-e)}, \boldsymbol{\theta}_e = x|\mathbf{a})}{P(\boldsymbol{\theta}^{(-e)}|\mathbf{a})} = \frac{\prod_{f=1}^{F} x_f^{\alpha_f + c_{e,f} - 1}}{B(\boldsymbol{\alpha}_e + \mathbf{c}_e)} \tag{7}$$

where $c_{e,f}$ is the number of times that word e is aligned to word f given $\mathbf{a}$, $\mathbf{s}$ and $\mathbf{t}$. Equation (7) is a consequence of the fact that the Dirichlet distribution is a conjugate prior to the categorical distribution, so that if

$$\mathbf{x} \sim \mathrm{Dir}(\boldsymbol{\alpha})$$
$$z \sim \mathrm{Cat}(\mathbf{x})$$

then given a sequence z of $|z|$ samples from $\mathrm{Cat}(x)$ we have

$$x|z \sim \mathrm{Dir}(\alpha + c(z)) \tag{8}$$

where

$$c(z)_m = \sum_{i=1}^{|z|} \delta_{z_i=m}$$

is the number of samples in z that are equal to m. This can be easily shown from the definition of the Dirichlet distribution using Bayes' theorem:

$$P(x|\alpha, z) \propto P(z|\alpha, x)P(\alpha, x) \tag{9}$$

$$\propto \prod_{i=1}^{d} x^{\alpha_i - 1} \prod_{i=1}^{|z|} x_{z_i} \tag{10}$$

$$= \prod_{i=1}^{d} x^{\alpha_i - 1} \prod_{m} x^{c(z)_m} \tag{11}$$

$$= \prod_{i=1}^{d} x^{\alpha_i + c(z) - 1} \tag{12}$$

which is the (unnormalized) Dirichlet distribution with parameter $\alpha + c(z)$.

Equation (6) and Equation (7) can be used for sampling with standard algorithms for categorical and Dirichlet distributions, respectively, and together they define an *explicit* Gibbs sampler for the Bayesian IBM model 1. While simple, this sampler suffers from poor mixing (Östling, 2015, section 3.3) and is not a competitive algorithm for word alignment. However, much better performance can be achieved by using a *collapsed* sampler where the parameters θ_e are integrated out so that we only have to derive a sampling equation for the alignment variables $P(a_j = i|a^{(-j)})$.

First we use Equation (5) to derive an expression for $P(a|s, t, \alpha)$, from which the final sampler can be computed as

$$P(a_j = i|a^{(-j)}, s, t, \alpha) = \frac{P(a^{(-j)}, a_j = i|s, t, \alpha)}{P(a^{(-j)}|s, t, \alpha)} \tag{13}$$

Since the elements of a are exchangeable, a sufficient statistic for a is the count vector $c(\cdot)$ where each element

$$c(a, e, f)_{e,f} = \sum_{k=1}^{K} \sum_{j=1}^{J^{(k)}} \delta_{s^{(k)}_{a^{(k)}_j} = e \wedge t^{(k)}_j = f} \tag{14}$$

represents the number of times that source word type e is aligned to target word type f under the alignment $\mathbf{a}$. Next, we marginalize over each of the lexical distributions θ_e.

$$P(\mathbf{a}|\mathbf{s},\mathbf{t},\boldsymbol{\alpha}) = \prod_{e=1}^{E} \int_{\Delta} P(\mathbf{a}_{\{j|s_{a_j}=e\}}|\theta_e,\mathbf{s},\mathbf{t},\boldsymbol{\alpha}) P(\theta_e|\mathbf{s},\mathbf{t},\boldsymbol{\alpha})\, d\theta_e \tag{15}$$

Substituting from Equation (5) into the integral we have

$$P(\mathbf{a}|\mathbf{s},\mathbf{t},\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{e=1}^{E} \int_{\Delta} \prod_{f=1}^{F} \theta_{e,f}^{c(\mathbf{a},\mathbf{s},\mathbf{t})_{e,f}+\alpha_f-1}\, d\theta_e \tag{16}$$

where the innermost product can be recognized as an unnormalized $\mathrm{Dir}(\boldsymbol{\alpha}+\mathbf{c}(\mathbf{a},\mathbf{s},\mathbf{t}))$ distribution which has normalization factor $B(\boldsymbol{\alpha}+\mathbf{c}(\mathbf{a},\mathbf{s},\mathbf{t}))$, so that the final expression becomes

$$P(\mathbf{a}|\mathbf{s},\mathbf{t},\boldsymbol{\alpha}) = \prod_{e=1}^{E} \frac{B(\boldsymbol{\alpha}+\mathbf{c}(\mathbf{a},\mathbf{s},\mathbf{t}))}{B(\boldsymbol{\alpha})} \tag{17}$$

$$= \prod_{e=1}^{E} \frac{\Gamma(\sum_{f=1}^{F}\alpha_f)\prod_f \Gamma(\alpha_f + c(\mathbf{a},\mathbf{s},\mathbf{t})_{e,f})}{\Gamma(\sum_{f=1}^{F}(\alpha_f + c(\mathbf{a},\mathbf{s},\mathbf{t})_{e,f}))\prod_f \Gamma(\alpha_f)} \tag{18}$$

Combining Equation (13) with Equation (18) gives us an expression where almost all of the terms are cancelled out, except when $s_i = e$ and $t_j = f$ for which $c(\mathbf{a},\mathbf{s},\mathbf{t})_{e,f}$ and $c(\mathbf{a}^{(-j)},\mathbf{s},\mathbf{t})_{e,f}$ differ by 1. We are left with a remarkably simple sampling distribution:

$$P(a_j = i|\mathbf{a}^{(-j)},\mathbf{s},\mathbf{t},\boldsymbol{\alpha}) = \frac{\alpha_{t_j} + c(\mathbf{a}^{(-j)},\mathbf{s},\mathbf{t})_{s_i,t_j}}{\sum_{f=1}^{F}\left(\alpha_f + c(\mathbf{a}^{(-j)},\mathbf{s},\mathbf{t})_{s_i,f}\right)} \tag{19}$$

By repeatedly sampling each a_j in turn from Equation (19) we are guaranteed to, in the limit, obtain an unbiased sample from $P(\mathbf{a})$ under the model. What we are really interested in, however, is to estimate the marginal distributions $P(a_j = i)$ as closely as possible while using as little computation as possible, given a sequence of correlated samples $\mathbf{a}^{(t)}$ for time $t \in 1 \ldots T$. Given a sequence of samples $\mathbf{a}^{(t)}$ we can then approximate the marginal distributions

$$P(a_j = i) = \mathop{\mathbb{E}}_{P(\mathbf{a})}[\delta_{a_j=i}] = \sum_{t=1}^{\infty} \delta_{a_j^{(t)}=i} \approx \frac{1}{T}\sum_{t=1}^{T} \delta_{a_j^{(t)}=i} \tag{20}$$

In practice $\mathbf{a}^{(0)}$ will be initialized either from a uniform distribution or by using the output of a simpler model, and the samples will gradually become more independent of $\mathbf{a}^{(0)}$ as t increases. Since $\mathbf{a}^{(0)}$ is likely to lie in a low-probability region of the model,

so do the initial samples, and it is common to use a *burn-in* period and disregard all $\mathbf{a}^{(t)}$ for $t < t_0$. To further ameliorate the problem of initialization bias, it is possible to run several independently initialized samplers and average their results. Combining these methods the marginal distribution approximation becomes

$$P(a_j = i) \approx \frac{1}{N(T - t_0 + 1)} \sum_{n=1}^{N} \sum_{t=t_0}^{T} \delta_{a_j^{(n,t)}=i} \tag{21}$$

where N is the number of independent samplers and t_0 is the length of the burn-in period. Finally, a better estimate can be obtained by applying the Rao-Blackwell theorem (Blackwell, 1947; Gelfand and Smith, 1991), which allows us to re-use the computations of $P(a_j = i|\mathbf{a}^{(-j)})$ during sampling and averaging these distributions rather than $\delta_{a_j^{(t)}=i}$. The final approximation then becomes

$$P(a_j = i) \approx \frac{1}{N(T - t_0 + 1)} \sum_{n=1}^{N} \sum_{t=t_0}^{T} P(a_j^{(n,t)} = i|\mathbf{a}^{(n,t)(-j)}) \tag{22}$$

3. Methods

We now turn to the particular models and algorithms implemented in EFMARAL, presenting our Bayesian HMM model with fertility, the Gibbs sampler used as well as the details on how to make it computationally efficient.

3.1. Alignment model

Our goal in this work is to find a word alignment algorithm that is both accurate and efficient. Previous studies have shown that good word order and fertility models are essential to high accuracy (Brown et al., 1993; Och and Ney, 2003), along with reasonable priors on the parameters (Mermer and Saraçlar, 2011; Östling, 2015). As was discussed in Section 2.3, MCMC algorithms and in particular collapsed Gibbs sampling are particularly suitable for inference in this class of models, as long as the convergence of the Markov chain are sufficiently fast. Even within this class of algorithms there are some trade-offs between accuracy and computational efficiency. In particular, hierarchical priors have been shown to somewhat improve accuracy (Östling, 2015, p. 65), but in spite of improved sampling algorithms (Blunsom et al., 2009) it is still considerably more costly to sample from models with hierarchical priors than with Dirichlet priors.

For these reasons, we use a HMM model for word order based on Vogel et al. (1996) as well as a simple fertility model, and the complete probability of an alignment is essentially the same as Equation (5) with extra factors added for the word order and

fertility model:

$$\begin{aligned} &P(\mathbf{s}, \mathbf{t}, \mathbf{a}, \theta, \psi, \pi, \alpha, \beta, \gamma) \\ &\quad \propto \left(\prod_{k=1}^{K} \prod_{j=1}^{J^{(k)}} \theta_{s_{a_j^{(k)}}^{(k)}, t_j^{(k)}} \right) \cdot \left(\prod_{e=1}^{E} \prod_{f=1}^{F} \theta_{e,f}^{\alpha_f - 1} \right) \\ &\quad \cdot \left(\prod_{k=1}^{K} \prod_{j=1}^{J^{(k)}+1} \psi_{a_j^{(k)} - a_{j-1}^{(k)}} \right) \cdot \left(\prod_{m=m_{min}}^{m_{max}} \psi_m^{\beta_m - 1} \right) \\ &\quad \cdot \left(\prod_{k=1}^{K} \prod_{i=1}^{I^{(k)}} \pi_{s_i^{(k)}, \phi(i, \mathbf{a}^{(k)})} \right) \cdot \left(\prod_{e=1}^{E} \prod_{n=0}^{n_{max}} \pi_{e,n}^{\gamma_n - 1} \right) \end{aligned} \tag{23}$$

where $\psi \sim \mathrm{Dir}(\beta)$ are the categorical distribution parameters for the word order model $P(a_j - a_{j-1} = m)$, and $\pi_e \sim \mathrm{Dir}(\gamma)$ for the fertility model $P(\phi(i, \mathbf{a}) | s_i = e)$. In our experiments we fix $\alpha = 0.001$, $\psi = 0.5$ and $\gamma = 1$, but these parameters are not very critical as long as $0 < \alpha \ll 1$.

The IBM models naturally allow unaligned source language words, but in order to also allow target words to not be aligned we use the extension of Och and Ney (2003) to the HMM alignment model, where each source word s_i (from sentence $\mathbf{s}$ of length I) is assumed to have a special NULL word s_{i+I}. The NULL word generates lexical items from the distribution θ_{NULL}, and the word order model is modified so that

$$P(a_j = i + I | a_{j-1} = i') = p_{\text{NULL}} \delta_{i=i'} \tag{24}$$

$$P(a_j = i + I | a_{j-1} = i' + I) = p_{\text{NULL}} \delta_{i=i'} \tag{25}$$

$$P(a_j = i | a_{j-1} = i' + I) = \psi_{i-i'} \tag{26}$$

where p_{NULL} is the prior probability of a NULL word alignment (fixed to 0.2 in our experiments).

We collapse the sampler over θ and ψ in the same manner as was shown in Section 2.4 and obtain the following approximate[2] sampling distribution:

$$\begin{aligned} P(a_j = i | \mathbf{a}^{(-j)}, \mathbf{s}, \mathbf{t}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \propto & \frac{\alpha_{t_j} + c(\mathbf{a}^{(-j)}, \mathbf{s}, \mathbf{t})_{s_i, t_j}}{\sum_{f=1}^{F} \left(\alpha_f + c(\mathbf{a}^{(-j)}, \mathbf{s}, \mathbf{t})_{s_i, f}\right)} \\ & \cdot \frac{\beta_{i - a_{j-1}} + c'(\mathbf{a}^{(-j)})_{i - a_{j-1}}}{\sum_{m=m_{min}}^{m_{max}} \left(\beta_m + c'(\mathbf{a}^{(-j)})_m\right)} \\ & \cdot \frac{\beta_{a_{j+1} - i} + c'(\mathbf{a}^{(-j)})_{a_{j-1} - i}}{\sum_{m=m_{min}}^{m_{max}} \left(\beta_m + c'(\mathbf{a}^{(-j)})_m\right)} \\ & \cdot \frac{\pi_{s_i, \phi(i, \mathbf{a}^{(-j)}) + 1}}{\pi_{s_i, \phi(i, \mathbf{a}^{(-j)})}} \end{aligned} \tag{27}$$

While collapsing over the θ is essential for acceptable mixing in the Markov chain, this is not the case for $\boldsymbol{\pi}$. Instead, we alternate between sampling from Equation (27) and

$$\boldsymbol{\pi}_e \sim \mathrm{Dir}\left(\boldsymbol{\gamma} + c''(\mathbf{a})_e\right) \tag{28}$$

where $c''(\mathbf{a})_e$ is the count vector over the fertility distribution for source word e given alignments $\mathbf{a}$. The advantage of this is that the last product of Equation (27) can be precomputed, saving computation in the inner loop in exchange for the (relatively minor) expense of also sampling from Equation (28).

3.2. Computational efficiency

From Equation (27) it is clear that the computational complexity of sampling sentence k is $O(I^{(k)} J^{(k)})$, since every alignment variable $a_j^{(k)}$ for each $j \in 1 \ldots J^{(k)}$ needs to evaluate the expression in 27 once for each $i \in 1 \ldots I^{(k)}$, and each evaluation requires constant time assuming that the sums are cached. Since sentence lengths are approximately proportional across languages, $I^{(k)} \approx \lambda J^{(k)}$ for some constant λ, this gives a total complexity of $O(\sum I^2)$ per iteration of sampling $\mathbf{a}$. Note that the complexity does not change as we go from Equation (19) for the simple IBM model 1 to Equation (27) for the more complex model with word order and fertility.

In contrast, the corresponding Expectation-Maximization (EM) algorithm for IBM alignment models has $O(\sum I^2)$ complexity in the E-step only for models with simple or no word order model. The HMM-based model of Vogel et al. (1996) can still be implemented relatively efficiently using dynamic programming, but complexity increases to $O(\sum I^3)$. For models with fertility computing the expectations instead becomes intractable, and previous authors have solved this by using approximative

[2]The approximation consists of ignoring the dependence between the two draws from the word order jump distribution (second and third factors).

greedy optimization techniques (Brown et al., 1993) or local Gibbs sampling (Zhao and Gildea, 2010). The main advantage of EM over a collapsed Gibbs sampler is that the former is trivial to parallelize, which makes well-implemented parallel EM-based implementations of simple alignment models with $O(\sum I^2)$ complexity, such as FAST_ALIGN (Dyer et al., 2013), a strong baseline performance-wise.

Algorithm 1 Inner loop of our sampler for IBM model 1

```
function SAMPLE(a_j^{(k)(-j)})
    ▷ Initialize cumulative probability
    s ← 0
    for all i ∈ 1 ... I^{(k)} do
        ▷ Load denominator reciprocal (small array random access)
        D^{-1} ← d_{k,i}
        ▷ Load numerator index (sequential access)
        L ← l_{k,i,j}
        ▷ Load numerator (large array random access)
        N ← u_L
        ▷ Compute unnormalized probability (one multiplication)
        p̂ ← D^{-1}U
        ▷ Accumulate probabilities (one addition)
        s ← s + p̂
        ▷ Store cumulative probability (sequential access)
        p_i ← s
    end for
    ▷ Sample from a uniform distribution on the unit interval
    r ~ Uniform(0, 1,)
    r ← r · p_I
    ▷ Find the lowest i such that p_i > r
    i ← 1
    while p_i ≤ r do
        i ← i + 1
    end while
    a_j^{(k)} ← i
end function
```

If a collapsed Gibbs sampler is to be a viable option for performance-critical applications, we must pay attention to details. In particular, we propose utilizing the fixed order of computations in order to avoid expensive lookups. Recall that variables $a_j^{(k)}$ are sampled in order, for $k = 1 \dots K$, $j = 1 \dots J^{(k)}$. Now, for each pair $\langle k, j \rangle$ we need

to compute

$$\frac{\alpha_{t_j} + c(\mathbf{a}^{(-j)}, \mathbf{s}, \mathbf{t})_{s_i,t_j}}{\sum_{f=1}^{F} \left(\alpha_f + c(\mathbf{a}^{(-j)}, \mathbf{s}, \mathbf{t})_{s_i,f}\right)}$$

which, if the numerator sum and the reciprocal of the denominator sum are cached in memory, involves two table lookups and one multiplication. Since multiplication is fast and the denominator reciprocal is stored in a relatively small dense array, most attention has to be paid to the numerator lookup, which apart from the constant α_{t_j} is a sparse matrix with non-zero counts $c(\mathbf{a}^{(-j)}, \mathbf{s}, \mathbf{t})_{s_i,t_j}$ only where s_i and t_j are aligned. The standard solution would therefore be to use a hash table with $\langle s_i, t_j \rangle$ as keys to ensure memory efficiency and constant-time lookup. However, most counts are in fact guaranteed to always be zero, as only words from the same parallel sentence pair can be aligned. We are therefore able to construct a count vector $\mathbf{u}$ and an index table $\mathbf{l}$ such that $u_{l_{k,i,j}} = c(\mathbf{a}^{(-j)}, \mathbf{s}, \mathbf{t})_{s_i,t_j} + \alpha_{t_j}$. At the expense of some extra memory usage we are able to achieve the lookup with only two operations, one of which is a cache-efficient sequential memory access. With this method, the inner loop of the sampler for IBM model 1 thus contains only six operations, outlined in algorithm 1. Adding the HMM word order model, two more sequential memory loads and two multiplications are needed, and adding the fertility model requires one more memory load and a multiplication.

4. Related work

In this section we relate our work mainly to the literature on Bayesian models of word alignment, as well as computationally efficient methods for this problem. A comprehensive survey of word alignment methods is beyond the scope of this article, for this we refer the reader to Tiedemann (2011).

Much of research into word alignment has been based on the pioneering work of Brown et al. (1993), and we have already introduced part of their family of IBM alignment models in Section 2.1. Their most advanced models still perform competitively after nearly two decades, but due to their complexity (with exact inference being intractable) many have suggested simpler alternatives, typically by keeping the lexical translation model intact and introducing computationally convenient word order and fertility models so that inference with the Expectation-Maximization (EM) algorithm remains tractable. Notable examples include the simple HMM-based model of Vogel et al. (1996) and the even simpler reparametrized IBM model 2 of Dyer et al. (2013). Neither of these include a model for word fertility, but Toutanova et al. (2002) showed that a simplified fertility model (which only counts alignments from consecutive target words) can be added to the HMM model without increasing the complexity of inference, and more recently this has also been achieved for a general fertility model (Quirk, 2013).

The EM algorithm requires computing the expected values of the alignments, $\mathbb{E}\left[\delta_{a_j=i}\right]$, given the current values of the model parameters. The authors cited above all dealt with this fact by analytically deriving expressions for exact computation of these expectations in their models. Zhao and Gildea (2010) instead chose to use Gibbs sampling to approximate these expectations, which allowed them to perform efficient inference with EM for a HMM model with fertility. Riley and Gildea (2012) later showed how Variational Bayesian techniques can be used to incorporate priors on the parameters of the IBM models, with only minor modifications to the expressions for the alignment expectations.

Recently, several authors have disposed with EM altogether, relying entirely on Gibbs sampling for inference in IBM-based models with Bayesian priors of varying complexity (Mermer and Saraçlar, 2011; Mermer et al., 2013; Gal and Blunsom, 2013; Östling, 2015). Of these, Gal and Blunsom (2013) and to some extent Östling (2015) prioritize maximizing alignment accuracy, which is obtained by using complex hierarchical models. Mermer et al. (2013) use Dirichlet priors with IBM models 1 and 2 to obtain efficient samplers, which they implement in an approximate fashion (where dependencies between variables are ignored during sampling) in order to facilitate parallelization. This article follows previous work by the first author (Östling, 2015), which however was focused on alignment of short parallel text for applications in language typology and transfer learning, rather than efficient large-scale alignment for use with statistical machine translation systems.

5. Results

In this section we first investigate the effect of different parameter settings in EFMARAL, then we proceed with a comparison to two other influential word alignment systems with respect to the performance of statistical machine translation (SMT) systems using the alignments. Since computational efficiency is an important objective with EFMARAL, we report runtime for all experiments.

The following three systems are used in our comparison:

GIZA++: The standard pipeline of IBM models with standard settings of 5 iterations of IBM 1, 5 iterations of the HMM model, and 5 iterations of IBM model 3 and 4 with Viterbi alignments of the final model (Och and Ney, 2003). Class dependencies in the final distortion model use automatically created word clusters using the MKCLS tool, 50 per language.

FAST_ALIGN: An log-linear reparameterization of IBM model 2 using efficient inference procedures and parameter estimations (Dyer et al., 2013). We use the options that favor monotonic alignment points including the optimization procedures that estimate how close they should be to the monotonic diagonal.

EFMARAL: Our implementation of the MCMC alignment approach proposed in this article.

Since these tools all use asymmetric models, we ran each aligner in both directions and applied the GROW-DIAG-FINAL-AND (Section 5.1) or GROW-DIAG-FINAL (Section 5.2) symmetrization heuristic (Och and Ney, 2003, p. 33). This method assumes a set of binary alignments, so for EFMARAL we produce these by choosing the single most probable value for each a_j: $\arg\max_i P(a_j = i)$. In this way the results are more easily comparable to other systems, although some information is lost before the symmetrization step and methods have been explored that avoid this (Matusov et al., 2004; Östling, 2015, pp. 46–47).

5.1. Alignment quality experiments

As discussed in Section 2.4, there are two ways of trading off computing time for approximation accuracy: increasing the number of independent samplers, and increasing the number of sampling iterations. Here we explore the effects of these trade-offs on alignment accuracy.

Following Och and Ney (2003), most subsequent research has compared the results of automatic word alignment to hand-annotated data consisting of two sets of links: S, containing *sure* tuples $\langle i, j\rangle$ where the human judgment is that s_i and t_j must be aligned, and $S \subseteq P$, containing *possible* tuples $\langle i, j\rangle$ where s_i and t_j may be linked. Given a set A of alignments to be evaluated, they define the measures precision (p), recall (r), and alignment error rate (AER) as follows:

$$p = \frac{|A \cap P|}{|A|} \tag{29}$$

$$r = \frac{|A \cap S|}{|P|} \tag{30}$$

$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{31}$$

While popular, the AER measure is biased towards precision rather than recall and correlates poorly with machine translation performance. Fraser and Marcu (2007) instead suggest to use the F-measure, which favors a balance between precision and recall as defined in Equation (29) and Equation (30):

$$F_\alpha = \left(\frac{\alpha}{p} + \frac{1-\alpha}{r}\right)^{-1} \tag{32}$$

In our experiments, we report both AER and $F_{0.5}$.

In order to evaluate alignment quality we are limited to language pairs with annotated alignment data. For this reason, we use the corpora and test sets from the WPT 2003 and 2005 shared tasks (Mihalcea and Pedersen, 2003; Martin et al., 2005). In addition, we also use the Swedish-English part of the Europarl corpus version 7

Table 1. Data sets used for our alignment quality experiments. The total number of sentences in the respective corpora are given along with the number of sentences and gold-standard (S)ure and (P)ossible alignment links in the corresponding test set.

Corpus	Sentences	Sentences	|S|	|P|
	Training	Test		
English-French	1,130,588	447	4,038	17,438
English-Romanian	48,641	200	5,034	5,034
English-Inuktitut	333,185	75	293	1,972
English-Hindi	3,556	90	1,409	1,409
English-Swedish	692,662	192	3,340	4,577

(Koehn, 2005) with test set from Holmqvist and Ahrenberg (2011). The data sets are presented in Table 1, where it can be noted they differ both in size and in annotation style. In particular, the English-Romanian and English-Hindi data only have one set of gold-standard links, so that $S = P$, the English-French and English-Inuktitut data have $|S| \ll |P|$, while the English-Swedish data lies somewhere in between.

Table 2: Results of our alignment quality experiments. All timing and accuracy figures use means from five independently initialized runs. Note that lower is better for AER, higher is better for $F_{0.5}$. All experiments are run on a system with two Intel Xeon E5645 CPUs running at 2.4 GHz, in total 12 physical (24 virtual) cores.

Configuration	Quality		Time (seconds)	
	AER	$F_{0.5}$	CPU	Wall
English-French				
FAST_ALIGN	15.3	86.2	4,124	243
1x iterations, 2 samplers	8.2	92.3	741	270
4x iterations, 2 samplers	8.1	92.2	2,700	809
16x iterations, 2 samplers	8.1	92.1	10,557	2,945
1x iterations, 1 samplers	9.1	91.4	470	248
1x iterations, 4 samplers	7.8	92.6	1,324	298
1x iterations, 8 samplers	**7.6**	**92.9**	2,456	330

Continued on next page

Configuration	AER	$F_{0.5}$	CPU	Wall
English-Hindi				
FAST_ALIGN	67.3	32.7	27	2
1x iterations, 2 samplers	48.3	51.7	107	12
4x iterations, 2 samplers	49.0	51.0	416	46
16x iterations, 2 samplers	51.0	49.0	1,664	183
1x iterations, 1 samplers	49.4	50.6	81	10
1x iterations, 4 samplers	47.5	52.5	146	13
1x iterations, 8 samplers	**46.7**	**53.3**	238	17
English-Inuktitut				
FAST_ALIGN	28.7	78.1	752	48
1x iterations, 2 samplers	22.3	81.5	160	62
4x iterations, 2 samplers	19.7	83.7	560	199
16x iterations, 2 samplers	**17.3**	**86.0**	2,176	747
1x iterations, 1 samplers	23.8	80.1	98	56
1x iterations, 4 samplers	19.6	84.1	259	64
1x iterations, 8 samplers	18.4	85.3	515	72
English-Romanian				
FAST_ALIGN	32.5	67.5	266	17
1x iterations, 2 samplers	28.7	71.3	167	47
4x iterations, 2 samplers	29.0	71.0	648	173
16x iterations, 2 samplers	29.5	70.5	2,580	682
1x iterations, 1 samplers	29.8	70.2	97	43
1x iterations, 4 samplers	28.2	71.8	320	53
1x iterations, 8 samplers	**27.9**	**72.1**	656	59
English-Swedish				
FAST_ALIGN	20.5	79.8	12,298	671
1x iterations, 2 samplers	13.1	87.0	1,606	589
4x iterations, 2 samplers	11.4	88.6	5,989	1,830
16x iterations, 2 samplers	**10.6**	**89.4**	23,099	6,519
1x iterations, 1 samplers	13.8	86.3	1,005	538
1x iterations, 4 samplers	13.2	86.8	2,681	626
1x iterations, 8 samplers	11.7	88.3	6,147	839

Table 2 shows the result of varying the number of samplers and iterations for all the language pairs under consideration. As a baseline for each language pair, we use FAST_ALIGN as well as the default EFMARAL configuration of two independent samplers, running $x = \lfloor 100/\sqrt{K} \rfloor$ sampling iterations where K is the number of parallel sentences in the data (with the additional constraint that $4 \leq x \leq 250$). Following

the practice set by Brown et al. (1993), each model is initialized with the output of a simpler model. For the full HMM+fertility model, we run $\lfloor x/4 \rfloor$ sampling iterations of IBM model 1 initialized with uniformly random alignments, use the last sample to initialize the fertility-less HMM model that we also run for $\lfloor x/4 \rfloor$ iterations. Finally, x samples are drawn from the full model and the final alignments are estimated from these using Equation (22).

The experiments described in Table 2 were carried out on a system with dual Intel Xeon E5645 CPUs, with a total of 24 virtual cores available. Even though this setup strongly favors FAST_ALIGN's parallel implementation, EFMARAL is faster for the largest corpus (where speed matters most) in terms of both wall time and CPU time, and for all but the smallest corpora in CPU time. This trend will also be seen in Section 5.2, where even larger parallel corpora are used for our machine translation experiments.

As expected, increasing the number of independently initialized samplers consistently results in better alignments, in line with research on model averaging for a wide range of machine learning models. When it comes to increasing the number of sampling iterations the result is less clear: for some pairs this seems even more important than the number of independent samplers, whereas for other pairs the quality metrics actually change for the worse. Recall that the samplers are initialized with a sample from the fertility-less HMM model, and that the correlation to this sample decreases as the number of samples from the HMM model with fertility increases. Decreasing quality therefore indicates that for that particular language pair and annotation style, the fertility model performs worse than the mix between the fertility and fertility-less models obtained by using a small number of samples. When interpreting these results, it is also important to keep in mind that the quality metrics are computed using discretized and symmetrized alignments, which are related in a quite complex way to the probability estimates of the underlying model.

From a practical point of view, one should also consider that additional independent samplers can be run in parallel, unlike additional sampling iterations which have a serial dependence. For this reason and because of the consistent improvements demonstrated in Table 2, increasing the number of samplers should be the preferred method for improving alignment quality at the cost of memory and CPU time.

5.2. Machine translation experiments

In order to test the effect of word alignment in a downstream task, we conducted some experiments with generic phrase-based machine translation. Our models are based on the Moses pipeline (Koehn et al., 2007) with data coming from the Workshop on Statistical Machine Translation. In our setup we use the news translation task from 2013 with translation models for English to Czech, German, Spanish, French and Russian and vice versa. Parallel training data comes from Europarl version 7 (Koehn, 2005) (for all language pairs except Russian-English) and the News Commentary corpus version 11. For language modeling, we use the monolingual data sets from Eu-

Table 3. Data used for training SMT models (all counts in millions). Parallel data sets refer to the bitexts aligned to English and their token counts include both languages.

	Monolingual		Parallel	
Language	Sentences	Tokens	Sentences	Tokens
Czech	8.4	145	0.8	41
German	23.1	425	2.1	114
English	17.3	411	–	–
Spanish	6.5	190	2.0	109
French	6.4	173	2.0	114
Russian	10.0	178	0.2	10

roparl and News Commentary as well as the shuffled news texts from 2012. We did not use any of the larger news data sets from more recent years to avoid possible overlaps with the 2013 test set. We apply a pipeline of pre-processing tools from the Moses package to prepare all data sets including punctuation normalization, tokenization, lowercasing and corpus cleaning (for parallel corpora). Statistics of the final data sets are listed in Table 3.

All language models use order five with modified Kneser-Ney smoothing and are estimated using KenLM (Heafield et al., 2013). Word alignments are symmetrized using the GROW-DIAG-FINAL heuristics and we use standard settings to extract phrases and to estimate translation probabilities and lexical weights. For reordering we use the default distance-based distortion penalty and parameters are tuned using MERT (Och, 2003) with 200-best lists.

Table 4 shows the performance of our SMT models given alignments from the different word alignment systems. The left-hand part of the table contains results when using full word forms for the word alignment systems, whereas the results in the right-hand part were obtained by removing any letters after the four first from each word, as a form of approximate stemming since all the languages in our evaluation are predominantly suffixing. Though seemingly very drastic, this method improves accuracy in most cases since data sparsity is a major problem for word alignment.

Next we turn to the computational cost of the experiments just described, these are found in Table 5. In almost all cases, EFMARAL runs faster by a comfortable margin. The only exception is for the smallest dataset, Russian-English, where FAST_ALIGN uses slightly less wall time (but still much more CPU time). This trend is also present in the alignment quality experiments in Section 5.1 with mostly smaller corpora, where EFMARAL is only faster for the largest corpus.[3]

[3]Due to different computing environments, only four CPU cores were available per aligner in the SMT experiments, versus 24 cores in the alignment quality experiments.

Table 4. Results from our SMT evaluation. The BLEU scores are the maximum over the Moses parameters explored for the given word alignment configuration.

	BLEU score					
Translation pair	No stemming			4-prefix stemming		
	EFMARAL	GIZA++	FAST_ALIGN	EFMARAL	GIZA++	FAST_ALIGN
Czech-English	**23.43**	23.29	22.77	**23.58**	23.57	23.44
English-Czech	**16.22**	15.97	15.69	**16.11**	15.96	15.88
German-English	23.60	**23.86**	22.84	23.54	**23.80**	23.08
English-German	**17.83**	17.69	17.50	**17.77**	17.70	17.65
Spanish-English	**28.50**	28.43	28.25	28.57	**28.69**	28.20
English-Spanish	27.39	**27.51**	27.08	**27.49**	**27.49**	27.08
French-English	**28.50**	28.45	28.06	**28.69**	28.67	28.33
English-French	**27.73**	27.57	27.22	27.66	**27.71**	27.16
Russian-English	**20.74**	20.14	19.55	**20.96**	20.65	20.38
English-Russian	**15.89**	15.55	15.07	**16.17**	16.13	15.77

Table 5. Timings from the word alignments for our SMT evaluation. The values are averaged over both alignment directions. For these experiments we used systems with 8-core Intel E5-2670 processors running at 2.6 GHz.

		Time (seconds)					
		Wall	CPU	Wall	CPU	Wall	CPU
Translation pair	Stem	EFMARAL		GIZA++		FAST_ALIGN	
Czech-English	no	**303**	**462**	13,089	13,083	465	1,759
Czech-English	yes	**233**	**361**	12,035	12,033	311	1,200
German-English	no	**511**	**766**	42,077	41,754	1,151	4,407
German-English	yes	**377**	**558**	43,048	43,023	813	3,115
Spanish-English	no	**500**	**782**	39,047	39,003	1,034	3,940
Spanish-English	yes	**346**	**525**	38,896	38,866	758	2,911
French-English	no	**696**	**1,088**	41,698	41,646	1,681	6,423
French-English	yes	**383**	**583**	40,986	40,907	805	3,101
Russian-English	no	122	**206**	3583	3581	**107**	382
Russian-English	yes	87	**151**	3148	3143	**78**	292

6. Concluding remarks

We hope that the reader at this point is convinced that Bayesian alignment models with Markov Chain Monte Carlo inference should be the method of choice for researchers who need to align large parallel corpora. To facilitate a practical shift towards this direction, we have released the EFMARAL tool which the evaluations in this article show to be both accurate, computationally efficient, and useful as a component of practical machine translation systems.

Acknowledgments

Computational resources for this project were provided by CSC, the Finnish IT Center for Science.[4]

Bibliography

Blackwell, David. Conditional Expectation and Unbiased Sequential Estimation. *The Annals of Mathematical Statistics*, 18(1):105–110, 03 1947. doi: 10.1214/aoms/1177730497. URL http://dx.doi.org/10.1214/aoms/1177730497.

Blunsom, Phil, Trevor Cohn, Sharon Goldwater, and Mark Johnson. A Note on the Implementation of Hierarchical Dirichlet Processes. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 337–340, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1667583.1667688.

Brown, Peter F., Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972470.972474.

Dempster, A. P., N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith. 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 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N13-1073.

Fraser, Alexander and Daniel Marcu. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*, 33(3):293–303, Sept. 2007. ISSN 0891-2017. doi: 10.1162/coli.2007.33.3.293. URL http://dx.doi.org/10.1162/coli.2007.33.3.293.

Gal, Yarin and Phil Blunsom. 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*, Stroudsburg, PA, USA, 2013. Association for Computational Linguistics.

[4] https://www.csc.fi/

Gelfand, Alan E. and Adrian F. M. Smith. Gibbs Sampling for Marginal Posterior Expectations. Technical report, Department of Statistics, Stanford University, 1991.

Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of ACL*, pages 690–696, 2013.

Holmqvist, Maria and Lars Ahrenberg. A Gold Standard for English-Swedish Word Alignment. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, number 11 in NEALT Proceedings Series, pages 106–113, 2011.

Koehn, Philipp. Europarl: A Parallel Corpus for Statistical Machine Translation. In *The Tenth Machine Translation Summit.*, Phuket, Thailand, 2005.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 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*, NAACL '03, pages 48–54, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073462. URL http://dx.doi.org/10.3115/1073445.1073462.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL*, pages 177–180, 2007.

Martin, Joel, Rada Mihalcea, and Ted Pedersen. Word Alignment for Languages with Scarce Resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, ParaText '05, pages 65–74, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1654449.1654460.

Matusov, Evgeny, Richard Zens, and Hermann Ney. Symmetric Word Alignments for Statistical Machine Translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1220355.1220387.

Mermer, Coşkun and Murat Saraçlar. 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*, HLT '11, pages 182–187, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-88-6. URL http://dl.acm.org/citation.cfm?id=2002736.2002775.

Mermer, Coşkun, Murat Saraçlar, and Ruhi Sarikaya. Improving Statistical Machine Translation Using Bayesian Word Alignment and Gibbs Sampling. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5):1090–1101, May 2013. ISSN 1558-7916. doi: 10.1109/TASL.2013.2244087.

Mihalcea, Rada and Ted Pedersen. An Evaluation Exercise for Word Alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond - Volume 3*, HLT-NAACL-PARALLEL '03, pages 1–10, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1118905.1118906. URL http://dx.doi.org/10.3115/1118905.1118906.

Och, Franz Josef. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*, pages 160–167, 2003.

Och, Franz Josef and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, Mar. 2003. ISSN 0891-2017. doi: 10.1162/089120103321337421. URL http://dx.doi.org/10.1162/089120103321337421.

Östling, Robert. *Bayesian Models for Multilingual Word Alignment*. PhD thesis, Stockholm University, 2015. URL http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-115541. ISBN 978-91-7649-151-5.

Pitman, Jim and Marc Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997. doi: 10.1214/aop/1024404422.

Quirk, Chris. Exact Maximum Inference for the Fertility Hidden Markov Model. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 7–11, 2013. URL http://aclweb.org/anthology/P/P13/P13-2002.pdf.

Riley, Darcey and Daniel Gildea. Improving the IBM Alignment Models Using Variational Bayes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 306–310, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390665.2390736.

Tiedemann, Jörg. *Bitext Alignment*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011.

Toutanova, Kristina, H. Tolga Ilhan, and Christopher Manning. Extensions to HMM-based Statistical Word Alignment Models. In *2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 87–94, 2002. URL http://ilpubs.stanford.edu:8090/557/.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 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, 1996. Association for Computational Linguistics. doi: 10.3115/993268.993313. URL http://dx.doi.org/10.3115/993268.993313.

Yarowsky, David, Grace Ngai, and Richard Wicentowski. Inducing Multilingual Text Analysis Tools via Robust Projection Across Aligned Corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1072133.1072187. URL http://dx.doi.org/10.3115/1072133.1072187.

Zhao, Shaojun and Daniel Gildea. 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, USA, October 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D10-1058.

Address for correspondence:
Robert Östling
robert.ostling@helsinki.fi
PL 24 (Unionsgatan 40, A316)
00014 Helsingfors universitet, Finland